

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

David Bavcon

**E-učni sistemi in priporočanje učnih
gradiv**

MAGISTRSKO DELO
ŠTUDIJSKI PROGRAM DRUGE STOPNJE
PEDAGOŠKO RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Zoran Bosnić

Ljubljana, 2017

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Zahvaljujem se Boštjanu Žnidaršiču in Gimnaziji Poljane, ki sta omogočila izvedbo testiranja sistema v razredu. Ravno tako se zahvaljujem mentorju izr. prof. dr. Zoranu Bosniću za pomoč in vse koristne nasvete pri izdelavi magistrskega dela.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Pregled področja	5
2.1	Učni sistemi	5
2.2	Učni objekti	6
2.3	Učna analitika	8
2.4	Pregled sorodnih del	10
3	Razvoj lastnega sistema za e-učenje	17
3.1	Orodja za razvoj in postavitev sistema	17
3.2	Podatkovni model	18
3.3	Opis sistema	22
3.4	Gradiva	32
4	Priporočilni sistem in rezultati	37
4.1	Cilj testiranja	42
4.2	Protokol testiranja	43
4.3	Pričakovanja	44
4.4	Potek poskusov	44
4.5	Rezultati	45
4.6	Pomen rezultatov	52

KAZALO

4.7	Primerjava z drugimi deli	53
5	Sklepne ugotovitve	59
5.1	Nadaljnje delo	61

Seznam uporabljenih kratic

kratica	angleško	slovensko
LMS	learning management system	sistem za upravljanje z učenjem
PMS	personal learning system	sistem za učenje posameznika
MOOC	massive open online course	množični odprti spletni tečaji
SCORM	sharable content object reference model	referenčni model za deljenje vsebine
ISMF	incremental simultaneous matrix factorization	inkrementalna hkratna matrična faktorizacija
LOM	learning object metadata	metapodatki učnih objektov

Povzetek

Naslov: E-učni sistemi in priporočanje učnih gradiv

V okviru našega dela smo razvili preprost e-učni sistem, ki priporoča gradiva dijakom. Najprej smo naredili kratek pregled o tem, kaj naj bi vsebovali e-učni sistemi. V nadaljevanju smo se lotili implementacije lastnega sistema, v katerem je na voljo tudi sprotna učna analitika. Za potrebe testiranja sistema smo naredili tudi nekaj gradiv, ki pokrivajo predmet informatika v gimnazijah. Potem smo v sistemu testirali tri različne načine priporočanja, in sicer: čim bolj podobno gradivo, čim bolj drugačno gradivo in naključni vrstni red gradiv. Vsakega od načinov je testirala tretjina dijakov. Na osnovi primerjave preciznosti, priklica in mere F1 smo ugotovili, da sta najboljša načina naključni in, takoj za njim, matrična faktorizacija, ki vrača medseboj čim bolj drugačno gradivo. Način, ki dijaku ponudi čim bolj medseboj podobno gradivo, se je v poskusih izkazal kot slab. Rezultati kažejo, da moramo učencu priporočiti naključno gradivo. Glede na preciznost z ostalimi načini priporočanja dosežemo slabše rezultate.

Ključne besede: e-učni sistemi, priporočanje gradiv, učna analitika.

Abstract

Title: E-learning systems and recommending learning materials

Within our thesis, we developed a simple e-learning system which recommends learning materials to high school students. We began by preparing a short overview of appropriate e-curriculums. We continued by implementing our own system that includes real-time learning analytics. To test the system, we also created some learning materials on high school information science. We tested three different recommendation techniques, i.e. learning materials that are the closest match to previous materials and the furthest match from previous materials and materials in random order. Each material assignment strategy was tested on one third of students. Based on observing precision, recall and F1 measure, we established that the best system turns out to be random order material assignment, followed by matrix factorisation that offers the least matching materials to previous materials. The technique offering the closest matching materials proved to be the least useful. The results therefore indicate that we should offer students random materials. Other strategies provided considerably worse results when we compared them by precision.

Keywords: e-learning system, recommendation of learning material, learning analytics.

Poglavje 1

Uvod

V šole vse bolj prodirajo razni e-učni sistemi. To poteka počasi, a uporaba se vsekakor povečuje. Tako se tudi vse bolj spreminja način učenja. Učenci se že učijo določene stvari po internetu ali uporabljajo različne učne sisteme, ker to morajo ali si želijo. Nekateri sistemi lahko služijo tudi zgolj za vajo in ponavljanje določene učne snovi. V Sloveniji je zelo prisoten sistem Moodle, ki ga uporabljajo tako osnovne in srednje šole kot fakultete.

Pojav MOOC-ov (angl. Massive Open Online Course) spreminja način učenja tako, da to poteka na daljavo. Sicer je res, da MOOC-i niso namenjeni ravno osnovnim in srednjim šolam, vseeno pa prispevajo k temu, da se e-učni sistemi in učenje na daljavo še bolj pospešeno uporabljajo ter da se e-učni sistemi še bolj razvijajo. V tako obsežnih e-učnih sistemih se zbere tudi veliko več podatkov o uporabnikih, zaradi česar se pričakujeta še boljša personalizacija in napredek na področju učne analitike [1].

E-učni sistemi so zelo različni. Nekateri so bolj ali manj namenjeni samo učenju, drugi pa skrbijo tudi za komunikacijo z uporabniki, imajo na voljo forum, komunikacijo med uporabniki, načine za skupinsko učenje in še bi lahko naštevali [2]. E-učni sistem je lahko zelo obsežen in ima na voljo veliko funkcionalnosti, v praksi običajno veliko funkcionalnosti, ki jih ponujajo različni sistemi, ne potrebujemo. V okviru magistrskega dela bomo zato naredili preprost sistem z osnovnim naborom najbolj potrebnih funkcionalnosti.

Pri izdelavi sistema ne moremo mimo učnih objektov, pod katere spadajo posamezna gradiva. Poskrbeti je treba, da lahko gradiva dokaj enostavno naredimo oziroma jih nekako uvozimo iz drugih sistemov [3]. Veliko težavo na področju učnih sistemov predstavljajo hiter razvoj tehnologije ter potrebe po stalnih popravkih in nadgradnjah sistema, da ne postane tehnološko zastarel [4]. Poleg tega se za učenje vse bolj uporabljajo tudi mobilne naprave, zato je treba poskrbeti tudi za ustrezno podporo.

V e-učnih sistemih so po navadi gradiva razvrščena tako, kot jih razvrsti učitelj, in v tem vrstnem redu jih učenci predelujejo. Učni sistemi pri tem ne izvajajo nobenega prilagajanja, personalizacije ali razvrščanja gradiv po pomembnosti. Vsako gradivo ni primerno za vse, saj lahko posameznega učenca določena vsebina ne zanima ali pa jo je že osvojil. Zaradi tega želimo učencu priporočiti gradivo, ki je zanj čim bolj primerno ali ki je vsebinsko zelo podobno tistemu, pri katerem se je slabo odrezal, da tako nadoknadi, česar ni ustrezno predelal. Hkrati želimo preizkusiti tudi različne načine in ugotoviti, kateri je najboljši za priporočanje gradiv. Na ta način nadgradimo bolj klasične e-učne sisteme. Pri načinu priporočanja gradiv imamo veliko možnosti. Gradiva so lahko razvrščena zgolj po številu ogledov in že to predstavlja neki način priporočanja, saj lahko slaba gradiva tako ostajajo bolj na dnu. Potem so tu še veliko bolj napredni načini z različnimi razdaljami (npr.: evklidska, kosinusna ...), v sklop še bolj naprednih načinov priporočanja pa spada matrična faktorizacija [1, 5, 6].

Sistem je treba preizkusiti z določeno vsebino, in ker v šolah računalnike uporabljajo večinoma samo pri informatiki, smo se odločili, da pripravimo gradiva, ki pokrivajo ta predmet. Tako bomo preizkusili različne načine priporočanja in ugotovili, kateri je boljši.

V primeru, ko učenci uporabljajo učni sistem v razredu, je še vedno treba določene stvari razložiti, sploh če delajo vsem težave. Še boljše je, če lahko to učitelj že sproti zazna in vnaprej opozori ostale, ki bodo verjetno imeli podobne težave. Učitelj je v šoli še vedno prisoten tudi, če učenci delajo samostojno. Če dobi podatke, da učenec nečesa ne razume, mu lahko to sproti

razloži. Za pomoč učiteljem pri pregledu uspešnosti učencev bomo v sistem vključili tudi učno analitiko, s pomočjo katere želimo imeti sproten vpogled v napredek učencev. Učna analitika je v zadnjem času zelo popularna, ker na različne grafične načine prikazuje napredek, rezultate in še marsikaj drugega.

Tako e-učni sistemi kot tudi učna analitika se bodo še naprej razvijali tudi v skladu s tehnološkim napredkom. Uporaba v šolah ali doma (npr. v okviru domače naloge) se bo v prihodnosti vsekakor bistveno povečala.

V delu bomo najprej na kratko pregledali področje. V naslednjem poglavju bomo opisali razvoj našega e-učnega sistema in njegovo delovanje. Na koncu bomo opisali priporočilni sistem in analizirali rezultate različnih strategij priporočanja učnih gradiv.

Poglavje 2

Pregled področja

Z razvojem tehnologije so se razvijali tudi e-učni sistemi, ki so bili sprva bolj preprosti in bistveno bolj omejeni v primerjavi s sistemi, ki jih poznamo danes. Sprva so bili e-učni sistemi preproste spletne strani z učnimi gradivi, saj se tehnologija še ni toliko razvila ali pa ni bila dovolj dostopna. Poleg tega so imeli tako šole kot učenci težave z dostopom do interneta. Z vse večjo dostopnostjo do interneta in razvojem tehnologij so s časom vključevali vse več funkcionalnosti (npr.: klepet, forum, načine za preverjanje znanja ...) in se počasi razvijali do te mere, da so postali adaptivni in se na različne načine prilagajajo uporabniku. Pri prilagajanju je zato včasih treba vložiti veliko več dela v pripravo gradiv, če je recimo treba pripraviti gradiva za različne učne sloge. V nadaljevanju bomo na kratko predstavili pomembne stvari, ki so povezane z e-učnimi sistemi.

2.1 Učni sistemi

Najbolj poznan in uporabljen učni sistem (LMS) v Sloveniji je Moodle, ki ga uporabljajo tako v osnovnih in srednjih šolah kot tudi na fakultetah. Poleg njega obstajajo tudi številni drugi sistemi, tako odprtokodni kot komercialni (npr. Blackboard) [7]. Različni sistemi imajo na voljo pestre nabore funkcionalnosti, vendar se v našem delu ne bomo posvečali že obstoječim

sistemom. V okviru učnih sistemov imamo možnosti za objavljanje gradiv, forume, možnosti za ocenjevanje izdelkov ... [2] Omeniti moramo še PLS (angl. Personal Learning System), ki so uporabniku prilagojena učna okolja. Poleg Moodla se v Sloveniji v manjši meri uporablja tudi Mahara, ki pa je bolj namenjena predstavitvi posameznega učenca in povezovanju z ostalimi učenci v razredu [8]. Odločili smo se, da naredimo lasten učni sistem, ki sicer ne bo imel toliko funkcionalnosti, kot jih ima Moodle, temveč bo bolj poenostavljen in prilagojen našim potrebam. Podrobnejši opis delovanja našega sistema sledi v nadaljevanju dela.

Pri uporabi e-učnih sistemov v šolah je velika težava dostop do interneta. Če imajo učenci neomejen dostop do interneta, namreč takoj odprejo še katero drugo spletno stran, ki ni v povezavi s trenutno učno snovjo. Enako velja za delo doma: če delajo za računalnikom, jih lahko takoj kaj zmoti in se lotijo česa drugega [2].

Gradiva v našem e-učnem sistemu so s področja računalništva in informatike. To ne pomeni, da je sistem omejen na to področje, saj bi lahko v njem imeli gradiva z vsebinsko različnih področij. V zvezi s tem trčimo na določene omejitve, recimo, da sistem ni posebej prilagojen za matematiko, ker nima možnosti, da bi pri nalogah sam izračunal vrednost izraza. Težava je lahko tudi vnos različnih matematičnih formul.

2.2 Učni objekti

V našem sistemu bodo tudi gradiva in tako ne moremo iti mimo učnih objektov. Učni objekti so v bistvu majhni kosi gradiv. Učni objekt je lahko zelo obsežen ali pa zelo drobnozrnat. V zvezi s tem pride do težave ponovne uporabnosti, ker bolj kot je učni objekt obsežen, težje ga ponovno uporabimo. Poleg tega je za učne objekte zaželeno, da so prenosljivi med različnimi učnimi sistemi in da jih ni treba v vsakem sistemu ustvariti znova. Obstajajo tudi različni standardi (npr. LOM, SCORM, IMS), ki se lahko uporabijo za definiranje in prenosljivost učnih objektov [9].

Predvsem je pomembno, da naš sistem omogoča tudi ponovno uporabo določenega gradiva, saj se določene stvari pokrivajo in bi jih lahko na ta način še enkrat uporabili. Hkrati lahko služi tudi kot dodatna utrditev znanja, če dobi učenec še enkrat isto gradivo ali zgolj njegov del. Druga pomembna stvar je, da sistem učiteljem poleg možnosti, da lahko ustvarjajo gradiva, omogoča tudi njihov izvoz. Na ta način privarčujemo na času, saj nam za drugi sistem gradiv ni treba ustvarjati znova. V okviru našega dela bomo tako poskusili narediti tudi izvoz gradiv v sistem Moodle, ki podpira naslednje standarde za uvoz gradiv: SCORM, AICC in IMS [10].

Omeniti moramo tudi standard SCORM (angl. Sharable Content Object Reference Model) [3], s katerim se učni objekti združijo v pakete, ki so potem prenosljivi med različnimi učnimi sistemi. Moodle podpira tudi standard SCORM. Ta standard pa je za nas zelo težko uporaben, ker hoče imeti zelo strukturirano strukturo in je popolnoma statičen. Je pa eden od možnih načinov za prenos gradiv iz našega sistema. V njegovem okviru se gradivom določi tudi njihova statična sekvenca, kar pa za nas spet ni uporabno, če želimo pot prilagajati.

Druga težava teh standardov je, da so bili narejeni pred leti, ko so bile spletne strani veliko bolj statične. Danes je že vsak sistem zelo dinamičen in standardi za današnje potrebe so preveč tehnološko zastareli. Sami standardi ne predvidevajo, da bi bil sistem bolj adaptiven in prilagajal pot posameznemu učencu. Ravno tako v okviru standardov ni poskrbljeno za prilagajanje glede na učencev učni slog. Sicer je to možno rešiti tako, da pripravimo za vsak slog svoje gradivo. Problem je tudi v tem, kako narediti vsebino bolj dinamično (npr. prikaz postopka po korakih, prikaz naključne naloge in ne vedno iste ...) v samem gradivu in da bo hkrati še vse delovalo v različnih sistemih. Mogoča rešitev bi bila, da bi naredili določene obdelave pri izmenjavi med sistemi. Narejene so bile tudi pobude za spremembe ali razširitve standardov, da bi bili bolj uporabni v današnjem času, a nekega širše uporabljenega standarda ni [3, 11].

V delu se bomo pri gradivih osredotočili zgolj na besedilo in slike, po za-

ključku sklopa pa bo sledila naloga, ki bo zgolj izbira med pravilnimi odgovori. Gradiva bodo organizirana najprej po predmetih, za primer, če bi sistem kdaj širili še na druga področja. Potem bodo organizirana v tematske sklope. Razmišljali smo tudi, da bi v okviru gradiv poskušali narediti možnost prilagajanja oziroma razbijanja gradiva v recimo dva dela za učenca, ki mu gre slabše; učenec, ki mu gre boljše, bi dobil gradivo v enem delu. Vendar se za to možnost nismo odločili, ker nismo imeli tako zelo obsežnih gradiv.

2.3 Učna analitika

V zadnjih letih je zelo popularna tudi učna analitika, ki je nastala zaradi vse večje razširjenosti in tehnološkega napredka e-učnih sistemov [1].

V okviru učne analitike lahko zbiramo številne informacije, kot so porabljen čas, število pravilnih rešitev, število napačnih odgovorov ... Zberemo lahko ogromno podatkov, težava pa je, kako jih ustrezno obdelati, uporabiti in prikazati. Določeni podatki niso najbolj zanimivi za učitelja. Težava je tudi, katere podatke bomo učitelju prikazali, da bo iz njih izvedel čim več. Če mu bomo namreč prikazali preveč podatkov, se ne bo znašel. Če želimo vse prikazati na eni strani v velikosti zaslona, smo poleg tega tudi prostorsko omejeni. Če pa ne bo vse na eni strani, učitelj med uro zagotovo ne bo gledal več strani hkrati, saj bi s tem izgubil preveč časa. Učna analitika naj bi bila prikazana kot nekakšna nadzorna plošča, na kateri imamo pregled nad vsem in s pomočjo katere dobimo vse koristne informacije na enem mestu. To naj bi bilo bistvo učne analitike, ostale stvari, kot so posamezni rezultati in primerjave, pa sodijo zgolj k statističnim podatkom, kjer pogledamo tisto, kar potrebujemo. Za učno analitiko je pomembno, da dobimo podatke v čim bolj realnem času, ne da analiziramo rezultate šele potem, ko so vsi zaključili, ali pa to naredimo kdaj drugič [12, 13].

Pri učni analitiki se vsi avtorji člankov veliko ukvarjajo z etiko. Če recimo preverimo, kdaj si učenec gradivo pogleda – ali je to počel ponoči itd. –, namreč lahko po eni strani posegamo v njegovo zasebnost. Podoben primer

je, če učitelj preveri, ali je učenec gradivo sploh pogledal, potem pa ga vpraša ravno nepredelano snov. Učna analitika je v člankih pogosto obravnavana kot priporočanje, in ne kot nekaj, kar ima na vpogled učitelj in iz česar sklepa, kako njegov razred napreduje [1].

V delu [14] so naredili sistem, kjer učitelj lahko spremlja, koliko zaostajajo učenci ali pa zgolj posamezni učenec za ostalimi. Tako ima narejeno vizualno časovnico, kjer vidi, kje so vsi učenci glede na čas. Omogočajo tudi pogled za učence, recimo koliko so ostali učenci porabili za neko dejavnost, tako da se lahko primerjajo med seboj. Zelo zanimiv je prikaz vseh učencev na časovnici in tega, kje je kdo. V našem sistemu imamo v zvezi s tem težavo, ker če želimo prilagajati pot, je mogoče, da si bo določeni učenec pogledal več gradiv in posledično porabil več časa. Čas je gotovo ena od spremenljivk, ki jo moramo pokazati. Druga stvar je tudi napredek razreda in stanje posameznega učenca v učnem procesu. V našem delu bomo naredili samo učno analitiko za učitelje, ne pa tudi za učence. Upoštevati moramo namreč dejstvo, da ni dobro, če bo določeni učenec, ki mu gre slabše, zmeraj na zadnjem mestu v primerjavi z ostalimi.

V drugem delu [12] lahko pride učitelj do veliko več podatkov, recimo kolikokrat so učenci dostopali do gradiv, njihove ocene po posameznih vprašanjih in število objav na forumu. Učitelj vidi, kako s časom vse bolj uporabljajo sistem.

V tem delu so sicer prikazane številne možnosti, ki jih nudijo v prikaz učitelju, vendar se v našem sistemu ne bomo toliko posvečali takim statistikam, temveč želimo učitelju ponuditi sistem, v katerem lahko med uro spremlja, kje so učenci.

Učna analitika je po navadi prikazana v okviru nadzorne plošče, kjer naj bi imel učitelj pregled nad vsemi pomembnimi podatki, seveda v okviru istega sistema. Ravno tako morajo biti vsi predstavljeni podatki enostavno razumljivi vsakomur. Težava učne analitike je tudi v tem, da to ni modul, ki bi ga lahko enostavno namestili v poljuben sistem, temveč je treba zadeve v vsakem sistemu implementirati na novo [12].

2.4 Pregled sorodnih del

Pregledali bomo sorodna dela, ki so se ukvarjala z različnimi načini priporočanja v e-učnih sistemih. Poudarili bomo tudi določene slabosti in prednosti, ki jih bomo nato upoštevali pri razvoju našega e-učnega sistema.

2.4.1 Pregled učnih sistemov

V večini omenjenih del v našem pregledu so se raziskovalci posvetili samemu priporočanju in niso podrobneje opisovali učnega sistema, ki so ga uporabili za testiranje. V delu [23] je učitelj lahko naredil gradivo iz obstoječih dokumentov (npr. prezentacij) tako, da je sistem sam dokument pretvoril v posamezne slike in jih vključil kot gradivo. S tem se je učitelj izognil ponovnemu ustvarjanju gradiv, ali kopiranju in lepljenju vsebin na posamezno prosojnico. Ko je učenec odprl gradivo, je imel prikazano prosojnico z navigacijo in menijem. Nad samo prosojnico je učenec lahko podal oceno gradiva in hkrati je imel prikazano tudi povprečno oceno tega gradiva. Pod samo prosojnico so bila prikazana druga gradiva, ki jih je priporočal sistem. Dobra stvar je vsekakor, da so poskrbeli, da ni potrebno gradiv delati na novo. Slabost tega načina je to, s slike ne moremo neposredno kopirati teksta. Problem je tudi, če bi želeli tekst na prosojnici popraviti, ker to pomeni, da je potrebno v originalnem gradivu narediti popravek in ga ponovno naložiti v sistem. V tem sistemu nas je zmotilo tudi to, da so priporočena gradiva prikazana že pod prosojnico, kar lahko povzroči, da učenec ne predela trenutnega gradiva, ampak gre že prej na drugega, ki se mu zdi bolj zanimiv.

Tutorski sistem so uporabljali v delu [21], ki je bil namenjen učenju programskega jezika Java. Prednost omenjenega sistema je, da se učenec v tem sistemu lahko več nauči in je vsekakor bolj personaliziran. Problem v tem sistemu je to, da ga težko uporabimo za različne druge vsebine in predmete.

Sistem, ki so ga uporabljali v delu [17] je bil sicer bolj namenjen preverjanjem znanja. Če je učenec rešil nalogo nepravilno je dobil namig, kako se nalogo reši pravilno. Poleg tega so imeli naloge, ki so bile sestavljene iz

dveh delov in pri katerih je bilo potrebno odgovoriti, ali trditev drži. Nato pa je bilo potrebno izbrati ustrezno razlago, zakaj trditev drži oziroma ne drži. Sistem je imel vsebine organizirane v različne sklope. Slabost tega je edino, da zahteva več dela pri ustvarjanju novih vsebin. Dobra stvar v tem sistemu so tudi vprašanja, ki so sestavljena iz dveh delov, ker mora tako učenec izbrati še pravo razlago.

2.4.2 Priporočanje v učnih sistemih

V različnih delih so se priporočanja gradiv in med gradivi lotili na podlagi ocen gradiv, statistike uporabe, pravil tipa “kaj če”, skupinskega izbiranja in še številnih drugih pristopov [1, 5, 6].

V delu [15] je opisan e-učni sistem, ki uporabniku priporoča gradiva glede na njegove prejšnje ocene v sistemu. Na začetku, ko uporabnik šele začne uporabljati sistem in nima o njem še nobene informacije, mu sistem predlaga gradiva s srednjo zahtevno stopnjo. Uporabnik na koncu gradiva dobi še dve vprašanji, ki sta povsod enaki. Pri prvem vprašanju uporabnik oceni težavnost gradiva na petstopenjski lestvici, pri drugemu vprašanju pa ocenjuje svojo sposobnost razumevanja gradiva. Na drugo vprašanje uporabnik odgovori zgolj z da ali ne. Sistem tako beleži več parametrov: težavnost gradiv, ki jih uporabnik pogleda, njegove sposobnosti, klike, iskanja in odgovore.

Za izračun novih ocen uporabljajo različne formule. Uspešnost priporočanja so merili na podlagi povprečja odgovorov na vprašanje (čim bolj kot so se približali 1.0, tem bolj so bili uspešni), če je uporabnik razumel vsebino. Poleg tega so na koncu uspešnost preverili še z anketo med uporabniki. Sistem omogoča tudi uporabo gradiv za neregistrirane uporabnike, vendar se v tem primeru priporočanje ne izvaja.

V opisanem sistemu nam ni vseh predvsem to, da mora uporabnik na koncu gradiva odgovoriti na vprašanji, ki ga zelo jasno sprašujeta o tem, kakšno zahtevnost ima posamezno gradivo. Na ta način je uporabniku takoj jasno, da bo, če bo kliknil odgovor, da je gradivo pretežko, dobil lažje gradivo. Poleg tega pride tudi do težave pristranosti uporabnikov, saj lahko

nekateri vedno ocenjujejo kot zelo dobro. Na podlagi tega smo se odločili, da bo v našem e-učnem sistemu na koncu le naloga, s pomočjo katere bomo ugotavljali, če je uporabnik razumel posamezno gradivo.

V delu [16] so za priporočanje uporabili preferenčno matriko, ki je zgrajena na podlagi zgodovine uporabnika in v kombinaciji z različnimi parametri. Rezultate na koncu s hibridnim načinom še izboljšajo. Parametre so računali po različnih formulah, ki so opisane v članku. Nanašale so se na najbolj uporabljena gradiva, podobna gradiva glede na uporabnika, podobna gradiva glede na podobne uporabnike in še na medsebojno kombinacijo teh parametrov. Kot zelo dobro so se izkazali skupinsko izbiranje (angl. collaborative filtering) in najbolj podobni ogledi gradiv v kombinaciji z najbolj podobnimi učenci.

Na podlagi članka ugotovimo, da zelo enostavne in pogosto uporabljene metode, kot je skupinsko izbiranje, dajejo zelo dobre rezultate.

Poleg priporočanja učnih gradiv za osvajanje nove snovi se Wang [17] ukvarja s priporočanjem vsebine za preizkuse znanja. Za nas to sicer ni najbolj uporabno, je pa v delu narejen širši pregled področja, v katerem je opisanih več možnosti pristopa prilagajanja poti. Pot prilagajajo na naslednje načine: s spreminjanjem vrstnega reda gradiv, skrivanjem določenih gradiv, dodajanjem komentarjev k povezavam in upoštevanjem predznanja, tako da se učencem z manj predznanja nudi več vodenja. V delu [18] je tudi pregled uporabe priporočilnih sistemov v e-učenju in e-učnih sistemih, njihovih implementacij in njihovih slabosti. Med slabostmi poudarjajo, da pristop ni bil ustrezno evalviran in da je bil preveč odvisen od ocen uporabnikov. V širšem pregledu [5] odkrivanja zakonitosti iz podatkov na pedagoškem področju so ugotovili, da je zelo pomembno, da odkrijemo učenčeve individualne posebnosti in se nanje ustrezno odzovemo v e-učnih sistemih, saj to zelo izboljša njegovo učenje. Srečamo se še z e-učnim sistemom [19], ki poleg priporočanja gradiv in celotne poti izvaja tudi funkcijo tutorskega sistema. Z vprašanji preverja osvojeno znanje, s pomočjo katerega nato uporabnika pouči o njegovih napakah.

Uporabljali so tudi pravila [20], na podlagi katerih so med drugim določili tudi učne sloge. Mi se za uporabo pravil nismo odločili.

Tako v našem sistemu kot tudi drugih sistemih je težava večinoma tudi z majhnim številom uporabnikov, kar lahko pripelje do tega, da priporočanje ni najbolj zanesljivo. Težavo subjektivnosti smo rešili tako, da uporabniki ne morejo gradiv neposredno ocenjevati, temveč to naredijo posredno, preko naloge. Še vedno pa obstaja možnost, da uporabnik namenoma zmeraj napačno odgovori, a tako ima bistveno manj vpliva.

V določenih sistemih so naredili tudi prilagajanje glede na posameznikov učni slog [21]. V delu [22] je omenjeno, da so za prilagajanje ucnim slogom gledali tudi uporabo foruma in klepeta. V številnih drugih delih so delali na podoben način, tako da niso gledali samo posameznih gradiv. Naš sistem posebnega prilagajanja ucnim slogom ne bo omogočal, tudi zaradi tega, ker bi za to morali narediti še bistveno kompleksnejši sistem, za kar bi potrebovali veliko več časa. Sistem bi potem moral podpirati še način, da bi po videoali avdioposnetku predvajal vsebino prosojnice.

V delu [23] so naredili priporočanje glede na vsebino in ocene uporabnikov. Uporabili so tri različne skupine za testiranje in vsaka skupina je testirala drugačen način. Ena skupina je uporabljala sistem brez priporočanja.

Ideja z različnimi skupinami je dobra za testiranje različnih načinov priporočanja. Za naš sistem se nismo odločili, da bi bila določena skupina brez priporočanja, temveč za to, da naj imajo določeni učenci naključno razvrščena gradiva.

Za zbiranje podatkov so uporabljali tudi dnevniške datoteke – tak način so uporabili v delu [24]. Priporočanje so izvedli na hibriden način, in sicer v kombinaciji med skupinskim izbiranjem ter na podlagi vsebine. Bolj zanimivo je to, da so upoštevali tudi zaporedje med gradivi (pot) in uporabili drseča okna z različnimi širinami, da so ugotovili, katera širina okna da boljši rezultat.

V našem delu bomo beležili zaporedje med gradivi, ki pa ga v priporočanju ne bomo uporabili.

Uspešnost so merili na različne načine, ki so bili odvisni od načina priporočanja. Za mere uspešnosti so tako uporabili preciznost, priklic in oceno F1. V večini del so uporabili samo preciznost in priklic, v nekateih delih pa poleg preciznosti še oceno F1. V primeru, da so uporabniki ocenjevali gradiva, je bila uporabljena ocena MAE (angl. Mean Absolute Error), ki vključuje predvideno oceno gradiva in oceno gradiva, ki jo poda uporabnik. Poleg nje so uporabljali še oceno ISM (angl. Intra-List Similarity Metric), ki pove uspešnost priporočanja [6]. V delih, kjer so uporabljali anketna vprašanja, so uporabljali razne statistične metode, na podlagi katerih so ocenjevali uspešnost dela učencev.

Ocenjevanje uspešnosti učne analitike v takšni implementaciji kot smo jo izvedli mi ni mogoče, ker nimamo ničesar uporabnega, kar bi lahko izmerili. Večinoma so uspešnost učne analitike ocenjevali z raznimi anketami ali opisno primerjavo z drugimi deli: kaj so izboljšali oziroma naredili novega v primerjavi z ostalimi deli [12, 25].

2.4.3 Slovenski prostor

Tudi v slovenskem prostoru je bilo že precej narejeno na področju e-učnih sistemov. Najbolj znan uporabljen je e-um (<http://www.e-um.si/>). V sistemu so gradiva za matematiko tako za osnovno kot tudi za srednjo šolo. Gradiva so razporejena po posameznih razredih in letnikih v skladu z učnim načrtom. Tudi učitelji so učencem zelo priporočali uporabo tega sistema za ponavljanje in utrjevanje svojega znanja. Zasnovan in narejen je bil že pred leti in danes ga pestijo predvsem težave zaradi tehnološkega razvoja (uporaba Appletov ni več zaželeno), zaradi česar je njegova raba za uporabnika zelo otežena. Sistem bi bilo treba v veliki meri predelati, kar pa terja veliko časa in tudi denarja. Ker je bil narejen pred leti, ne podpira naprednega priporočanja. V letih, ko je bil sistem narejen, je bil tehnološko primerljiv z ostalimi, ki so bili narejeni v tistem času. Sistem uporablja tudi standard SCORM [26].

V opisanem sistemu je vredno pozornosti predvsem to, kako so gradiva organizirana. Gradiva so organizirana po letnikih, kar pri matematiki ni

težava, saj jo imajo v srednji šoli vsi vsa štiri leta. Pri informatiki pa pride do težave zato, ker imajo dijaki predmet samo v prvem letniku in v četrtem letniku, če ga izberejo kot maturitetni predmet. Nekatere gimnazije imajo informatiko tudi v drugem letniku, ker imajo drugače razporejene ure predmeta po letnikih ali pa ker so se odločili za dodatne ure. Zaradi opisane težave v našem sistemu ne bomo razporejali gradiv fiksno po letnikih.

Tudi v Sloveniji so se ukvarjali z izbiro primernega učnega sistema in kakovostjo učnih gradiv. Opisane imamo tudi določene smernice, da je za učence, ki se recimo težko skoncentrirajo, dobro, da imajo na voljo samo golo besedilo brez vizualnih dodatkov, kar bomo upoštevali tudi v našem sistemu [26, 7].

Druga stvar, ki se jo zavedamo, je, da se tehnologije zelo hitro razvijajo in posledično bo tudi naš sistem kmalu zastarel ali zgolj nedelujoč ali pa ga bodo pestile podobne težave kot opisanega, tako da ga bo treba tehnološko posodobiti. Zaradi tega bomo v sistemu poskušali uporabljati čim manj tehnologij, ki bodo čim bolj splošno uporabljene [4]. Pozornost moramo nameniti tudi temu, da je treba vsake toliko časa vsebine posameznih gradiv prebrati, saj lahko postanejo zastarele.

V Sloveniji je bilo narejenih precej sistemov z učnimi gradivi ¹, nekatera zgolj za posamezen predmet, drugi pa za različne predmete. Sistemi so bolj ali manj osredotočeni na to, da gre uporabnik zaporedoma čez gradiva, tako da je vse bolj ali manj fiksno določeno. Gradiva imajo zraven še kakšno sliko ali videoposnetke. Preverjanje znanja je večinoma izpuščeno, ker bi to zahtevalo kompleksnejši sistem.

¹Nekaj primerov bolj preprostih sistemov: <http://www.egradiva.si/>, <http://egradiva.gis.si/>, <http://gradiva.txt.si/> ...

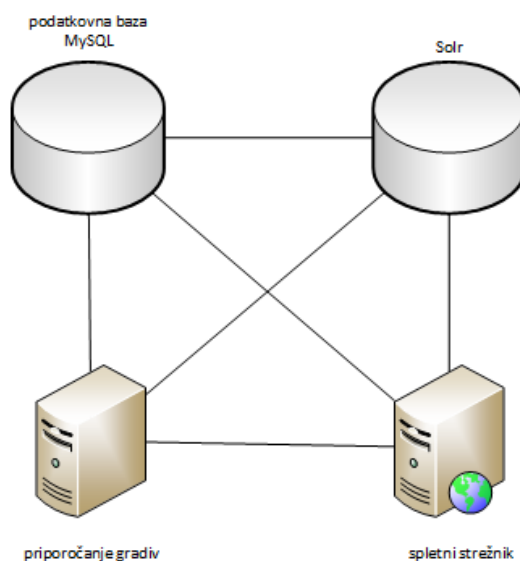
Poglavje 3

Razvoj lastnega sistema za e-učenje

3.1 Orodja za razvoj in postavitve sistema

E-učni sistem je bil razvit kot spletna aplikacija v programskem jeziku PHP. Del sistema, ki skrbi za priporočanje, je bil razvit v Pythonu (verziji 3). Za hitrejši razvoj spletne aplikacije smo uporabili ogrodje Symfony (verzijo 3.1.4). Za del sistema, ki skrbi za priporočanje, smo uporabili ogrodje Django REST framework (verzijo 3) in knjižnico scikit-learn. V spletni aplikaciji imamo na voljo tudi iskanje po gradivih, za kar je bila uporabljena iskalna platforma Apache Solr. Omenjena iskalna platforma omogoča iskanje po celotnem besedilu, označevanje zadetkov, pridobivanje golega besedila iz različnih tipov dokumentov (npr. pdf), fasetno iskanje (najpogosteje ga srečamo v spletnih trgovinah in nam omogoča filtriranje rezultatov iskanja na podlagi kriterijev npr. cena od 100 do 200 €) in še razne druge funkcionalnosti. Uporabljen je tudi spletni strežnik Apache.

Celoten sistem (slika 3.1) je sestavljen iz spletnega strežnika, algoritma za priporočanje gradiv, podatkovne baze MySQL in iskalne platforme Apache Solr. Vsi podatki so shranjeni v podatkovni bazi MySQL. Iz nje pridobivata podatke algoritem za priporočanje in spletni strežnik. Iskalna platforma Apa-



Slika 3.1: Komponente sistema, ki lahko tečejo na istem ali na več računalnikih

che Solr samo uvozi podatke in jih shrani. Iskalnik v spletni aplikaciji črpa podatke za funkcije samodokončaj, označene dele besedila, kjer je iskani niz iz iskalne platforme. Spletni strežnik pošlje še vse statične datoteke (npr.: .pdf, .doc ...), ki so lahko v našem sistemu samostojna gradiva. To naredimo, da v spletni aplikaciji omogočimo iskanje tudi po statičnih datotekah in ne samo gradivih, ki so narejena v naši spletni aplikaciji. Priporočilni algoritem uporablja iskalno platformo za to, da pridobi podobne dokumente glede na podani dokument. Na spletnem strežniku teče spletna aplikacija, ki komunicira z ostalimi prej omenjenimi deli sistema.

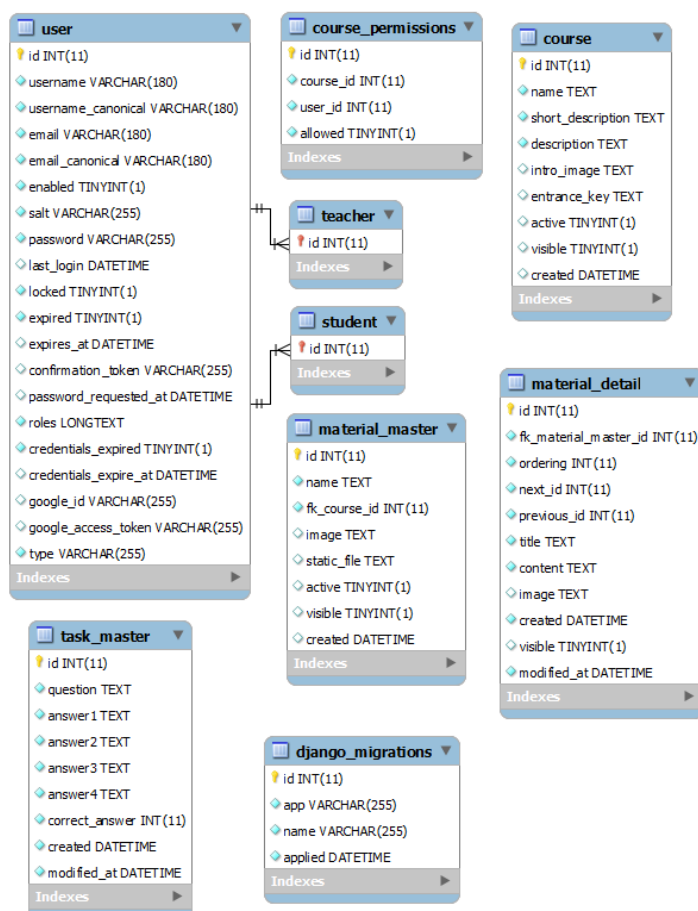
3.2 Podatkovni model

Podatkovni model obsega 14 tabel, ki jih lahko v grobem razdelimo na dva dela. Prvi del (slika 3.2) skrbi za delovanje sistema, uporabnike in vsebino, drugi del (slika 3.3) podatkovnega modela pa je namenjen beleženju podatkov medtem, ko se uporablja sistem. Najprej bomo začeli s prvim delom

podatkovnega modela in sicer v eni tabeli imamo vse podatke o uporabnikih (uporabniško ime, elektronski naslov ...). K uporabnikom spadata še dve tabeli: prva vsebuje identifikatorje uporabnikov, ki imajo vlogo študent, druga identifikatorje uporabnikov, ki imajo vlogo učitelj. Naslednja tabela vsebuje vse predmete v sistemu – tukaj imamo attribute: ime, kratek opis, opis, pot do slike, opcijski vhodni ključ, zastavico aktivno, zastavico vidno ter datum in čas nastanka predmeta.

Za nadzor dostopa do posameznih predmetov imamo tabelo, v kateri so shranjeni tuji ključ do predmeta, tuji ključ do uporabnika in atribut, ki pove, če ima uporabnik odobren ali zavrnjen dostop. Glavna tabela vsebuje gradiva – tukaj imamo parametre: ime, tuji ključ do predmeta, pot do slike, zastavice za aktivnost in vidnost gradiva ter datum in čas nastanka gradiva. H gradivom spada še podrejena tabela, kjer so shranjene posamezne prosojnice. V njej so tuji ključ na glavno tabelo z gradivom, vrstni red prosojnice, identifikator na prejšnjo in naslednjo prosojnico, naslov prosojnice, vsebina, zastavica za vidnost ter datum in čas nastanka prosojnice. Poleg tega imamo še atribut datum in čas spremembe, če je bila prosojnica kdaj spremenjena. Za naloge imamo svojo tabelo, kjer imamo vprašanje, štiri odgovore, številko pravilnega odgovora ter datum in čas nastanka ter morebitne spremembe. Tukaj bi lahko optimizirali tako, da bi imeli en pravičen odgovor in tri nepravilne odgovore, saj bi tako odpadla potreba po številki pravilnega odgovora.

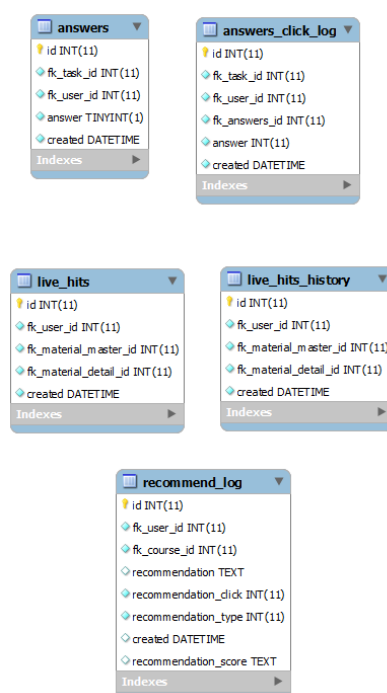
V nadaljevanju si bomo pogledali tabele, ki shranjujejo podatke, ko uporabnik uporablja sistem (slika 3.3). V tabeli, kjer shranjujemo narejena priporočila, imamo naslednje attribute: tuji ključ do uporabnika, tuji ključ do predmeta, priporočila kot besedilo (priporočila so ločena z vejico), na katero priporočilo je bilo kliknjeno, tip priporočanja, izračunane ocene priporočil do posameznih gradiv kot besedilo (ocene so ločene z vejico) ter še datum in čas priporočila. Sledita še dve tabeli, ki beležita obisk vsake prosojnice. V obeh imamo attribute: tuji ključ do uporabnika, tuji ključ do gradiva, tuji ključ do prosojnice ter še datum in čas obiska. Razlika med tabelama je samo v tem, da v prvi hranimo obiske vseh prosojnic, v drugi pa samo zadnjo obi-



Slika 3.2: Del podatkovnega modela, ki se nanaša na sistem. Relacije na sliki niso prikazane.

skano prosojnico. V primeru, da je uporabnik prišel do konca gradiva, bo zadnja prosojnica naloga. Naslednja tabela shranjuje odgovore pri nalogah. Tukaj imamo attribute: tuji ključ do naloge, tuji ključ do uporabnika, status odgovora pravilen ali pa nepravilen ter datum in čas nastanka. Menjava-nje odgovorov shranjujemo v ločeni tabeli z atributi: tuji ključ do naloge, tuji ključ do uporabnika, tuji ključ do odgovora, odgovor ter datum in čas nastanka.

Poleg omenjenih tabel imamo še sistemsko tabelo za migracije, ki jo potrebuje naše ogrodje.



Slika 3.3: Drugi del podatkovnega modela, ki beleži podatke med samo uporabo sistema. Te podatke potem uporabimo za priporočanje in pri prikazu učne analitike. Relacije na sliki niso prikazane.

Beleženje podatkov

V nadaljevanju so opisani podatki, ki jih beleži sistem o učencih. Pri vsaki nalogi se shranijo podatek o tem, ali je bil odgovor pravilen ali ne, datum in ura. Pri odgovorih se beleži tudi menjavanje odgovorov, preden učenec preveri rešitev. Tako se shrani podatek o tem, katere odgovore je izbral. Iz podatkov seštejemo število menjav odgovorov. Iz tega ugotavljamo negotovost uporabnika pri odgovorih. Za sledenje napredka po prosojnicah se shranjuje podatek o tem, na kateri prosojnici je kateri učenec. Zaradi hitrejšega prikazovanja smo se odločili, da napredek shranjujemo na dva načina: samo maksimalno doseženo prosojnico in vsako pogledano prosojnico posebej. Shranjujemo še podatke o gradivih, ki so bila priporočena, in o tem, na katero gradivo je uporabnik dejansko kliknil, ter način, na katerega je bilo

gradivo priporočeno.

3.3 Opis sistema

E-učni sistem je zasnovan za dve vlogi uporabnikov, in sicer uporabnika z vlogo učenec ter uporabnika z vlogo učitelj. V našem sistemu lahko vsak učitelj ustvari razred, v njem pa posamezna gradiva. Učitelj ni vezan na posamezni razred in lahko pregleduje tudi razrede, ki jih ustvarijo drugi učitelji. V primerjavi z drugimi naš sistem nima funkcionalnosti, kot so: forum, obveščanje, povezava s socialnim omrežjem, možnost za skupinsko delo ...

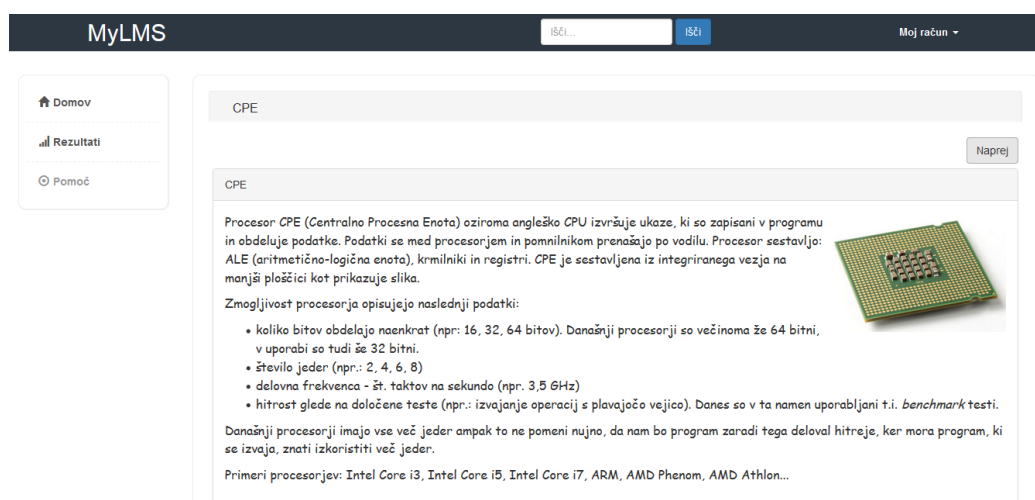
V nadaljevanju je predstavljen bolj podroben pogled glede na posamezno vlogo uporabnika.

3.3.1 Učenčev pogled

Uporabniku z vlogo učenec se po opravljeni prijavi oziroma registraciji (če sistema še ni uporabljal) prikažejo vsi razredi, ki obstajajo v sistemu. Učenci se v sistem lahko prijavijo tudi z Googlovim računom, kjer registracija ni potrebna. Če razred ni zaklenjen s ključem, lahko učenec vanj vstopi. V nasprotnem primeru mora ob prvem vstopu v razred vnesti pravi ključ, ki ga pove učitelj. Ko učenec vstopi v razred, se mu prikažejo vsa gradiva, ki niso skrita, pa tudi gradiva, ki niso aktivna. To pomeni, da gradivo sicer vidi, a ga še ne more odpreti. Za razliko od sistema Moodle gradiva niso razporejena po časovnici, temveč drug ob drugem. Za tak pogled smo se odločili, ker nismo želeli, da učenec pregleduje gradiva po določenem predpisanem redu. Drugi glavni razlog je, da bi težko priporočali gradiva, če bi imeli zgolj npr. tri gradiva za vsak teden. Gradiva so razvrščena v vrstnem redu, kot jih priporoči sistem za posameznega učenca.

Učenec tako vidi dva tipa gradiv, in sicer statične datoteke (npr. dokument v Wordu, PDF-ju ...) ter naš format gradiva. Učenec lahko tipa gradiv loči na podlagi ikone, ki se pojavi samo pri statičnih datotekah, medtem ko

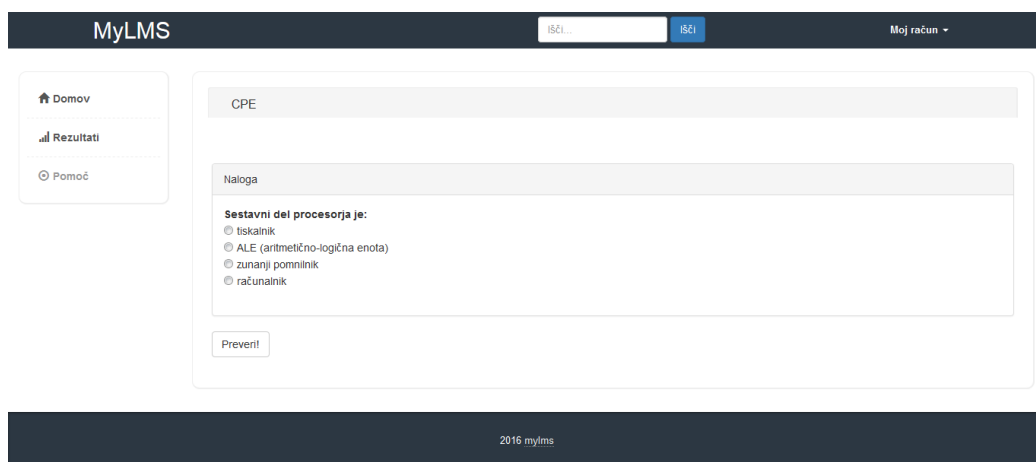
naš format gradiva nima nobene posebne oznake. Statična gradiva nimajo naloge, kar je posledica tehničnih omejitev. Naš format gradiv je sestavljen iz posameznih prosojnic, ki se pokažejo učencu in med katerimi se premika s klikom na gumb naprej ali nazaj. Potem, ko učenec pregleda vse prosojnice (slika 3.4), učenca pričaka naloga (slika 3.5), na katero mora ustrezno odgovoriti. Naloga ima vprašanje in vedno so na voljo štirje različni odgovori, med katerimi je eden pravilen, ostali pa nepravilni. Učenec lahko izbere samo en odgovor. Ob kliku na gumb preveri se prikaže sporočilo, kjer piše, ali je odgovoril pravilno ali napačno, poleg tega pa se posamezni odgovori označijo z barvo, tako da učenec vidi, kateri odgovor je pravilen (slika 3.6). To je zanj sprotna povratna informacija in tako je seznanjen tudi s tem, kateri odgovor je pravilen, v primeru, če je odgovoril napačno. Učenec ne dobi pojasnila, zakaj je določeni odgovor pravilen.



Slika 3.4: Primer posamezne prosojnice, ko učenec odpre gradivo.

Po končani nalogi se učenec vrne na stran, kjer so prikazana vsa gradiva, in sistem mu ponovno prikaže vsa gradiva v vrstnem redu, kot jih priporoča (slika 3.7). Razlika je le v tem, da so sedaj na koncu označena že obiskana gradiva.

Učenec lahko v sistemu pregleda svoje rezultate, in sicer vidi doseženi rezultat pri posamezni nalogi pa tudi, kdaj je nalogo opravil. Poleg pre-



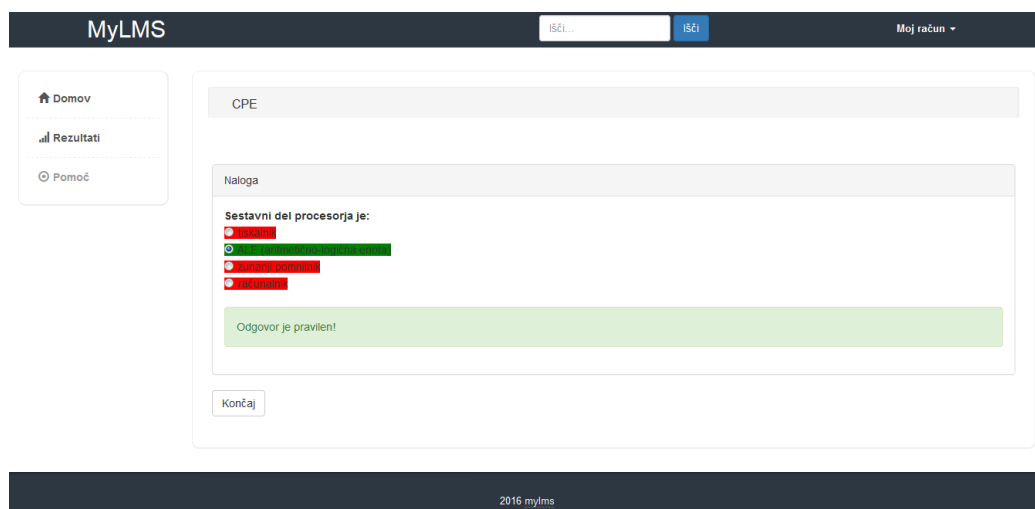
Slika 3.5: Primer naloge, ki je na koncu gradiva.

gleda rezultatov ima na voljo še iskanje, s katerim lahko najde gradivo, ki ga zanima.

Sistem nima nobene funkcionalnosti, ki bi služila obveščanju učencev, ravno tako nima na voljo foruma ali kakšnega drugega načina za medsebojno komunikacijo med učenci ali z učiteljem.

3.3.2 Pogled učitelja

Uporabniku z vlogo učitelj se po opravljeni prijavi oziroma registraciji (če prvič uporablja sistem) prikaže pogled z vsemi razredi, ki obstajajo v sistemu (slika 3.8). Učitelj lahko ureja obstoječe razrede ali ustvari novega. Pri ustvarjanju novega razreda mora podati podatke: ime razreda, kratek opis, opis, vstopni ključ, zastavici za aktivno in vidno. Vstopni ključ je opcijski, a z njim lahko učitelj omogoči dostop do razreda samo tistim učencem, ki imajo ključ. Zastavica vidno služi temu, da lahko razred skrijemo pred učenci. Podobno vlogo ima zastavica aktivno, ki je namenjen temu, da učenci razred sicer vidijo, ampak vanj ne morejo vstopiti, dokler ni aktiven. Smisel te zastavice je bolj v tem, da učenec vidi, kje je recimo razred, in da lahko učitelj to tudi pokaže, a jim še ne dovoli dostopa. Podatke o razredu lahko učitelj kadarkoli naknadno spreminja.



Slika 3.6: Sprotna povratna informacija učencu, ko zaključi z nalogo.



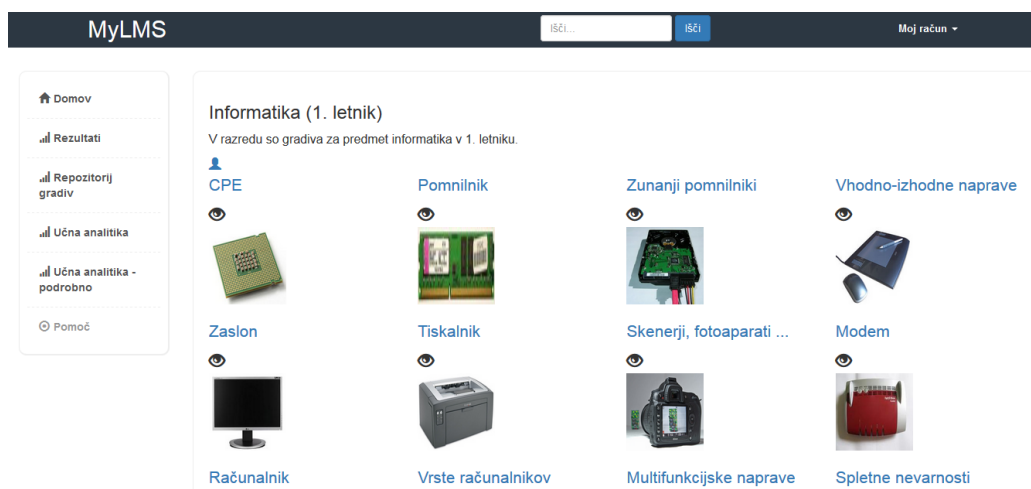
Slika 3.8: Začetna stran sistema po uspešni prijavi. Prikažejo se vsi razredi.

Ob vstopu v razred se odpre pogled z vsemi gradivi v njem (slika 3.9). Učitelj lahko ustvari novo gradivo, ki je lahko statična datoteka ali pa naš format gradiva. Dodajanje novega gradiva je enako za oba tipa gradiva, razlikuje se le v tem, da pri statičnem dodamo datoteko formata, npr.: .doc, .pdf, .xls ... Če želi učitelj ustvariti naš format gradiva, ne doda datoteke. Pri vsakem gradivu mora učitelj napisati tudi njegov naslov, dodati sliko ter še zastavici vidno in aktivno. V primeru, da je učitelj dodal zgolj statično datoteko, je z ustvarjanjem gradiva že končal, če pa je izbral naš format



Slika 3.7: Prikaz priporočenih gradiv učencu.

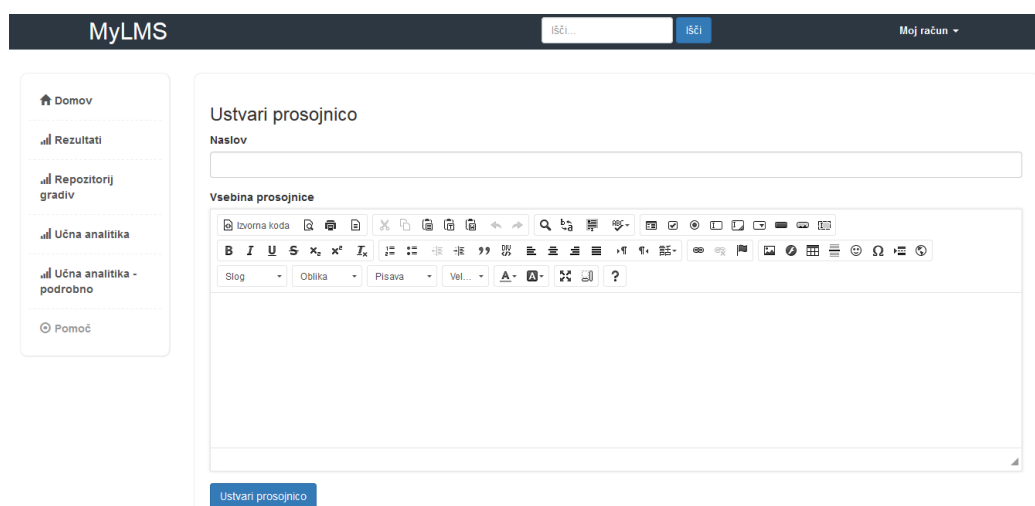
gradiva, ob kliku na gradivo šele začne z ustvarjanjem njegove vsebine.



Slika 3.9: Prikaz gradiv, ki so v razredu.

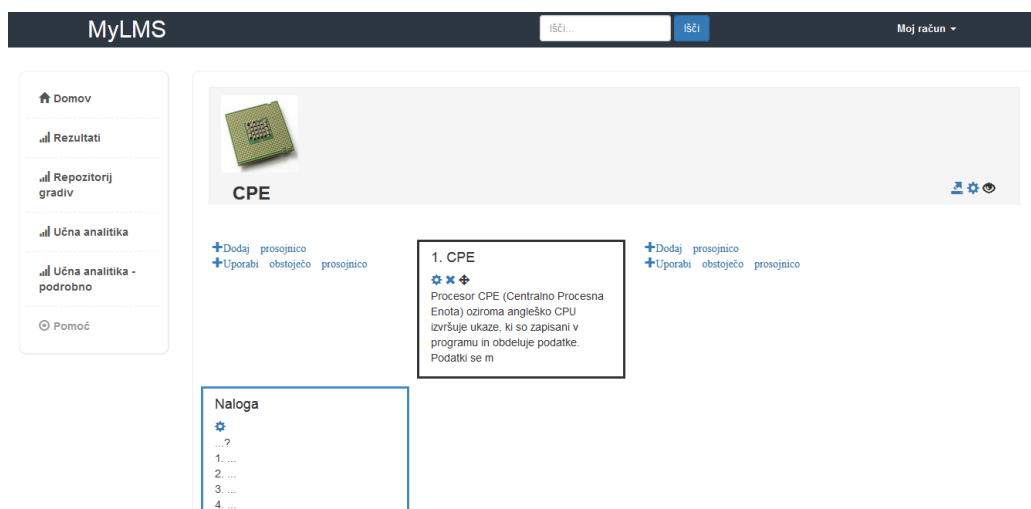
V našem formatu gradiva lahko učitelj ustvarja posamezne prosojnice in nalogo, ki sledi na koncu (slika 3.10). Za tak format gradiva smo se odločili zaradi pogostosti izdelave predstavitev v programu Powerpoint v izobraževanju. Da bi sistem čim lažje uporabljali, smo se morali čim bolj približati načinu dela v Powerpointu. Učitelj tako v naš format gradiva do-

daja posamezne prosojnice. Vsako prosojnico ustvari s pomočjo preprostega urejevalnika na način WYSIWYG. Urejevalnik vsebuje standardne funkcije, kot so: velikosti črk, pisava, slogi, poravnava besedila, vstavljanje tabel ... Prosojnico je mogoče urejati tudi v izvorni kodi HTML. Vsaki prosojnici je treba podati naslov. Enako deluje kopiranje in lepljenje besedila, le da naš urejevalnik ne zna ohraniti naprednega oblikovanja besedila, če ga kopiramo iz Wordovega dokumenta ali s spletne strani. V takih primerih se ohrani zgolj besedilo. Ko je prosojnica ustvarjena, se pokažejo vse prosojnice v trenutnem gradivu in njihova razporeditev. Sedaj je mogoče dodati novo prosojnico na mesto, ki ga izberemo s klikom na povezavo "dodaj prosojnico" (slika 3.11). Učitelj lahko tako doda poljubno število prosojnic.



Slika 3.10: Ustvarjanje nove prosojnice.

Za boljšo ponovno uporabo že narejene prosojnice v drugem gradivu ima učitelj možnost uporabe obstoječe prosojnice, ki jo lahko bodisi pusti takšno, kot je, ali pa spremeni. Učitelj lahko spreminja tudi vrstni red prosojnic s pomočjo načina povleci in spusti. K vsakemu gradivu doda še nalogo, ki je na koncu za zadnjo prosojnico. Pri nalogi napiše vprašanje in štiri možne odgovore, od katerih je eden pravilen, ostali trije pa so napačni. Odgovori se učencu pokažejo v naključnem vrstnem redu. Tako prosojnico in nalogo kot celotno gradivo lahko učitelj še naknadno popravlja in spreminja. Ravno



Slika 3.11: Urejanje gradiva.

tako lahko tudi briše posamezne prosojnice ali pa celotno gradivo.

Ker smo želeli implementirati ponovno uporabo gradiv, lahko učitelj dostopa do repozitorija vseh gradiv v sistemu, ne glede na razred in učitelja, ki je gradivo naredil. Tako lahko vsi učitelji v sistemu kopirajo gradiva v svoj razred in jih po potrebi spremenijo ali še izboljšajo; učitelj samo izbere, v kateri razred želi gradivo skopirati. Vsebina se pri tem skopira v novo gradivo, medtem ko originalno gradivo ostane nedotaknjeno.

Gradiva, ki so bila narejena v našem sistemu, je mogoče posamično tudi izvoziti tako, da se ob kliku na gumb generira datoteka PDF. Izvoz je sicer pomanjkljiv, ker ne ohrani vsega naprednega oblikovanja posamezne prosojnice, temveč zgolj besedilo in slike. Če bi želeli od izvozu ohraniti vso oblikovanje, bi nam to vzelo precej časa. Namenjen je bolj temu, da lahko učitelj narejeno gradivo v enem dokumentu dobi iz sistema, ne da bi vsako prosojnico posebej ročno kopiral recimo v program Word.

Učitelj lahko v sistemu pregleda tudi rezultate učencev po posameznih razredih, kjer vidi učenca, naslov gradiva, ki ga je obdelal, rezultat pri nalogi ter datum in uro, ko je z nalogo zaključil. Rezultat pri nalogah je lahko samo pravilen ali nepravilen, ker so naloge tako zasnovane, da učenec izbere samo en pravilen odgovor.

Učitelj lahko tudi odvzame dostop posameznemu učencu, da ne more več dostopati do razreda. Funkcionalnost deluje samo, če morajo učenci za vstop v razred vpisati ključ.

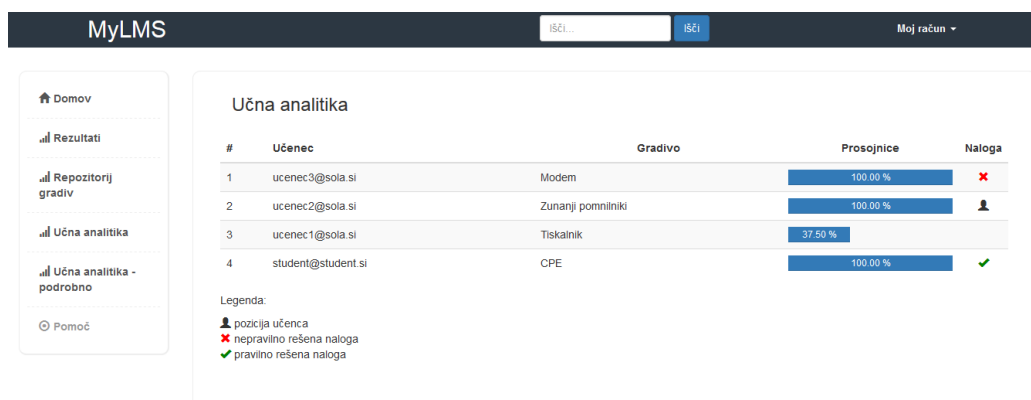
Sistem omogoča iskanje po gradivih, tako kot je to omogočeno učencem. Iskalnik omogoča tudi funkcijo samodokončaj in v rezultatih iskanja označi iskani niz.

Učna analitika

V okviru učne analitike sta pripravljena dva različna pogleda. Prvi je splošni (slika 3.12) in prikazuje napredek vseh učencev v sistemu v odstotkih ne glede na gradivo. Drugi pogled je osredotočen na napredek učencev pri izbranem gradivu (slika 3.15). Na ta način je učitelju omogočeno spremljanje napredka pri samostojnem delu celotnega razreda. Pri učni analitiki so prikazana samo gradiva v našem formatu, medtem ko statičnih datotek ne prikazujemo. Razlog za to je v tehnični izvedbi, saj ne moremo ugotoviti, na kateri strani je recimo učenec, če odpre Wordovo datoteko.

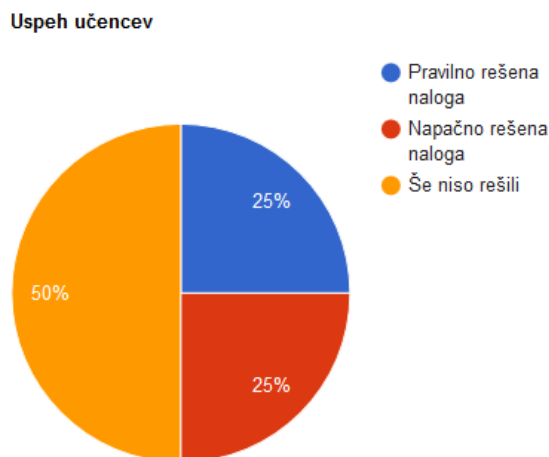
Pri prvem pogledu (splošnem pregledu vseh učencev v razredu) učitelj vidi učence, naslove gradiv, napredek po prosojnicah v odstotkih in stanje pri nalogi. Pogled se vsakih nekaj sekund avtomatsko osveži. Na tak način učitelj dejansko vidi, kako se spreminjajo odstotki pregledanih prosojnic, in tudi, kako hitro učenci napredujejo. Ko pride učenec do naloge, se na tistem mestu pojavi ikona za uporabnika. Ko nalogo zaključi, se v primeru pravilnega odgovora pojavi zelena kljukica, v primeru napačnega odgovora pa rdeč križec.

Pod to tabelo je še krožni diagram, ki prikazuje v odstotkih, koliko učencev je rešilo nalogo pravilno, koliko nepravilno ali koliko jih še ni rešilo naloge (slika 3.13). Zadnje lahko pomeni tudi, da prosojnice še pregledujejo in niso prišli do nalog. Naslednji graf prikazuje odstotek pravih odgovorov v primerjavi z nepravilnimi odgovori v zadnjih 24 urah. Iz njega lahko učitelj razbere, katera gradiva delajo največ težav učencem in kje potrebujejo še dodatno razlago, ali pa da je treba gradivo popraviti, da bo bolj razumljivo.

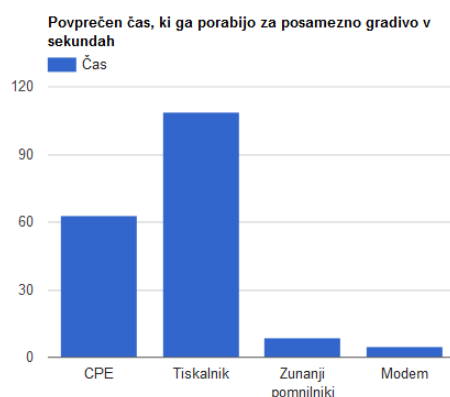


Slika 3.12: Vmesnik, ki učitelju omogoča spremljanje učencev.

Zadnji graf prikazuje povprečen čas (v sekundah), ki ga porabijo učenci, da pregledajo posamezno gradivo (slika 3.14). Iz njega učitelj zlahka razbere, koliko časa so učenci namenili posameznemu gradivu. Iz tega je tudi hitro razvidno, če učenci samo na hitro “preklikajo” gradivo, saj je v tem primeru porabljen čas zelo nizek.

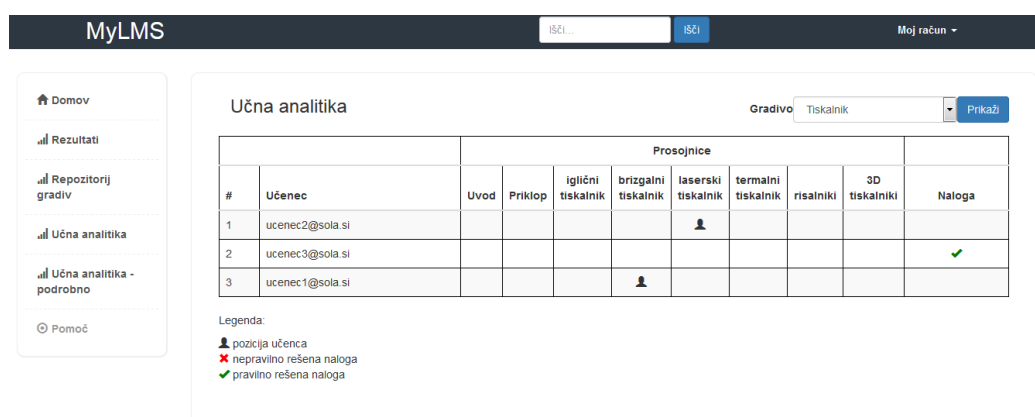


Slika 3.13: Prikaz uspeha učencev v okviru učne analitike.



Slika 3.14: Prikaz povprečnega časa, ki ga učenci porabijo za posamezno gradivo.

Drugi podroben pogled prikazuje učno analitiko glede na izbrano gradivo, ki ga izbere učitelj (slika 3.15). Za razliko od prejšnjega pogleda tukaj napredek ni predstavljen po prosojnicah, temveč točno na posamezno prosojnico. V tem pogledu učitelj točno vidi, do katere prosojnice so prišli učenci. Na koncu je tako kot v prejšnjem pogledu še stanje naloge. Pod tabelo je ravno tako kot v prejšnjem pogledu krožni diagram, ki v odstotkih prikazuje, koliko učencev je rešilo nalogo pravilno, koliko nepravilno in koliko jih še ni rešilo naloge.



Slika 3.15: Prikaz napredka učencev pri posameznem gradivu.

V našem delu imamo učno analitiko samo za učitelje, medtem ko bi lahko imeli tudi prikaz za učence, npr. kje so v primerjavi z drugimi učenci in kako napredujejo.

3.4 Gradiva

V našem sistemu so posamezna gradiva učni objekti, ki so različno obsežni (npr. samo ena prosojnica ali pa več prosojnic). Gradiva v sistemu so bila pripravljena za informatiko v prvem letniku gimnazije. Za četrti letnik jih nismo pripravili iz dveh razlogov. Prvi je premalo dijakov, ki bi preizkušalo naš sistem. Drugi razlog je povezan s prvim, in sicer da bi posledično morali dobiti veliko šol za sodelovanje, da bi dobili toliko dijakov, kot jih je na eni gimnaziji v prvem letniku. Tipično ima informatiko v četrtem letniku samo ena skupina dijakov na gimnaziji, kar pomeni največ 16 dijakov. Na gimnazijah nad sodelovanjem v četrth letnikih niso preveč navdušeni, ker se pripravljajo na maturo.

Odločili smo se, da bomo pripravili gradiva z vsebino, ki so jo učenci že predelali, in gradiva z vsebino, ki je še niso obravnavali. Glede vsebine smo se dogovorili s profesorjem na gimnaziji, kjer je potekalo testiranje. Pripravili smo gradiva iz učnih sklopov (po učnem načrtu za informatiko), kot so: osnove informatike, predstavitev informacij in digitalna tehnologija. Prva dva sklopa so učenci že predelali, tretjega še niso. Ravno tako je minilo že nekaj časa od takrat, ko so obravnavali prvi sklop, tako da so določene stvari že malo pozabili [27].

Iz tematskega sklopa *osnove informatike* so bila pripravljena gradiva za vsebine po učnem načrtu, kot so: temeljni pojmi, družbeni vidiki informatike in komuniciranje. To ne pomeni, da smo pokrili čisto vso snov, ki jo obsegajo vsebine, temveč večji del. Enako velja tudi za naslednja dva učna sklopa. Naslovi naših gradiv, ki so pokrivala ta sklop [27]:

- podatek, informacija,
- zvezno, diskretno,

- zaščita podatkov,
- komuniciranje,
- razmerja v komuniciranju,
- informacijska tehnologija.

Naslednji učni sklop *predstavitev informacij* je pokrival isto imensko vsebino. Naslovi gradiv, ki so pokrivala ta sklop:

- vrste informacij,
- številski sistemi,
- kodiranje.

Zadnji učni sklop *digitalna tehnologija* je pokrival vsebine iz učnega načrta, kot so: programska oprema računalnika, zgradba in delovanje računalnika ter strojna oprema računalnika. Naslovi gradiv, ki so pokrivala ta sklop:

- centralna procesna enota,
- pomnilnik,
- zunanji pomnilniki,
- vhodno-izhodne naprave,
- zaslon,
- tiskalnik,
- optični čitalci, fotoaparati ...,
- modem,
- računalnik,
- vrste računalnikov,

- multifunkcijske naprave,
- spletne nevarnosti,
- programska oprema,
- sistemska programska oprema,
- uporabniška programska oprema,
- avtorske zaščite,
- računalniški vsiljivci.

Določeno snov so že obdelali s profesorjem in bi morali vsebine že poznati. Gradiv, ki so povezana s strojno opremo, pa še niso predelali in na tem smo naredili tudi poudarek. Na voljo so imeli 26 gradiv.

Pri ustvarjanju smo v določena gradiva dali več slik, v nekatera pa sploh ne. Določena gradiva smo naredili bolj obsežna v primerjavi z ostalimi. Čeprav smo to naredili zaradi različnih učnih slogov, to ne pomeni, da smo pokrili vse učne sloge.

Vsako gradivo je imelo na koncu nalogo z vprašanjem in štirimi možnimi odgovori (slika 3.5), od katerih je eden pravilen, ostali trije pa nepravilni. Vprašanja so bila nižje zahtevnosti. Pri dveh nalogah je bilo treba izračunati pravilen odgovor. Vse, kar je bilo treba izračunati, je bilo množenje, za katerim je sledila izbira pravilnega odgovora. Po Bloomovi taksonomiji so bili vsa vprašanja in naloge na taksonomski ravni znanja in razumevanja. To sta sicer najnižji taksonomski stopnji, ampak če bi želeli pokriti še višje taksonomske stopnje, bi morali imeti tudi drugačne tipe nalog. Zaradi tega bi bila primerjava rezultatov težja. To bi morali v priporočanju nekako upoštevati in ustrezno nadgraditi sistem priporočanja. Tudi pri nalogah bi morali dobiti informacijo, na kateri taksonomski stopnji je katera.

Gradiva so po navadi odvisna drugo od drugega in snov se nadgrajuje. V našem primeru smo lahko naredili gradiva, ki niso bila odvisna od drugega gradiva. Če bi imeli gradiva, ki bi se navezovala, bi morali tudi ustrezno

prilagoditi priporočanje, da bi se priporočalo gradivo iz ustreznega sklopa. S tem bi imeli v vsakem sklopu precej manj gradiv in to bi precej vplivalo na priporočanje. Naš e-učni sistem bi morali ustrezno prilagoditi, da bi gradiva uvrstili v ustrezne sklope, oziroma nekako definirati, katera gradiva morajo biti obdelana predhodno. Tak način dela marsikomu ni všeč, saj ima vsak svoj načrt, kako bo obdelal učno snov. V tem primeru ima vsak učenec čisto drugačno zaporedje obdelovanja gradiv, kar je lahko težava, če učitelj želi nekaj razložiti, vsi pa tega še niso predelali.

Poglavje 4

Priporočilni sistem in rezultati

V našem e-učnem sistemu želimo učencem ponuditi gradiva, ki so za njih čim bolj primerna. Naš priporočilni sistem bo učencem priporočil na osnovi podatkov, ki jih bomo zbrali med samo uporabo sistema. Priporočanje gradiv bomo izvajali glede na besedilo posameznih gradiv v primeru napačno rešene naloge. V primeru pravilno rešene naloge pa bomo priporočali na osnovi podatkov o drugih uporabnikih in njihovih ocen gradiv. Uporabili bomo algoritem ISMF (inkrementalna hkratna matrična faktorizacija, angl. *incremental simultaneous matrix factorization*), medtem ko je za podobnost dokumentov glede na vsebino uporabljena komponenta “MoreLikeThis” iskalne platforme Apache Solr.

Odločili smo se, da bomo preizkusili različne načine priporočanja in ugotovili, kateri od njih je boljši. Prva težava pri snovanju priporočanja je bila, kaj bomo naredili na začetku, ko še nimamo nobenih podatkov o uporabniku. Odločili smo se, da na začetku pokažemo gradiva, ki so bila največkrat obiskana. Težava se pojavi le, če teh podatkov nimamo. To smo rešili z naključnim vrstnim redom gradiv. Pri priporočanju smo se odločili, da bomo upoštevali, če je bila naloga pri prejšnjem gradivu rešena pravilno, in se na podlagi tega odločili, kaj bomo naredili.

V primeru, da dijak naloge ne reši pravilno, mu skušamo poiskati podobno gradivo, da še malo utrdi snov. Če je naloga rešena pravilno, izvedemo ek-

speriment, in sicer ena tretjina dijakov dobiva čim bolj ustrezno gradivo na podlagi vseh zbranih podatkov do tistega trenutka. O uporabnikih imamo zbrane podatke o vseh gradivih, ki so jih predelali, in rezultate pri nalogah, ki nam predstavljajo oceno gradiva. Če uporabnik ni predelal še nobenega gradiva, potem o tem uporabniku še nimamo nobenih podatkov. V nadaljevanju se izvede priporočanje, glede na podatke o trenutnem uporabniku in še na podlagi podatkov o vseh ostalih uporabnikih. Seveda se priporoča na ta način samo, če je uporabnik že predelal kakšno gradivo, ker v nasprotnem primeru še nimamo podatkov o tem uporabniku. Druga tretjina dijakov dobi gradiva, priporočena ravno v obratnem vrstnem redu, in sicer če nekaj obvlada, mu damo nekaj novega, česar še ne pozna, da se bo nekaj novega naučil. Zadnja tretjina dijakov je imela gradiva vedno podana v naključnem vrstnem redu. Podrobnejša razlaga posameznega načina sledi v nadaljevanju.

Razlaga posameznih načinov priporočanja

Priporočilni sistem izvaja različna priporočanja glede na uspešnost reševanja zadnje naloge in razpoložljivost podatkov o uporabniku. Skrbi tudi za shranjevanje izbranega načina priporočanja in priporočila za kasnejšo analizo uspešnosti.

Priporoči gradiva v primeru pravilne naloge – v tem načinu se z uporabo matrične faktorizacije napove najprimernejše gradivo na podlagi podatkov o oceni gradiva in uporabnika. Namesto, da bi učencu omogočili ocenjevanje gradiv z ocenami od 1-5, smo ocene gradiv nadomestili s samodejnima ocenama 0 ali 1, glede na to, ali je učenec pravilno rešil nalogo. Ocena 0 predstavlja nepravilno, 1 pa pravilno rešeno nalogo. S tem smo učencu skrili ocenjevanje gradiv v primerjavi z drugimi deli, ki smo jih omenili pri pregledu. Ravno tako smo se s tem znebili tudi pristranskosti uporabnika, ker nekateri bi potem ocenjevali gradiva samo z oceno 4 ali 5, manj pa sploh ne. Nekateri uporabniki so lahko bolj kritični in bi zmeraj dali oceno 3. Pogosto se dogaja tudi, da na petstopenjski lestvici uporabniki zelo malokrat dajejo oceno 1 ali 2. Tako smo zagotovili bolj objektivno ocenjevanje gradiv, saj

Algoritem 1 algoritem za priporočanje poenostavljen

```

1: function PRIPOROČI(u_id, r_id):
2:   # u_id predstavlja identifikator uporabnika
3:   # r_id predstavlja identifikator razreda
4:   if uporabnik_že_predelal_vsa_j_eno_gradivo(u_id, r_id) then
5:     if zadnja_naloga_pravilna(u_id, r_id) then
6:       if u_id%3 == 0 then
7:         gradiva = priporoči_najbolj_podobno_gradivo(u_id, r_id)
8:       if u_id%3 == 1 then
9:         gradiva = priporoči_najmanj_podobno_gradivo(u_id, r_id)
10:      if u_id%3 == 2 then
11:        gradiva = priporoči_naključno_neobiskano_gradivo(u_id,
12:          r_id)
13:      else
14:        # nepravilna zadnja naloga - priporoči podobno gradivo glede
15:        # na samo vsebino zadnjega gradiva, ki ni bilo rešeno pravilno
16:        gradiva = priporoči_najbolj_podobno_gradivo(u_id, r_id)
17:      else
18:        # če nimamo podatkov o uporabniku, vrnemo največkrat obi-
19:        skana
20:        gradiva = najbolj_obiskana(r_id)
21:      # če nismo uspeli priporočiti nobenega gradiva, vrnemo naključna
22:      # gradiva izmed neobiskanih s strani podanega uporabnika
23:      if len(gradiva) == 0 then
24:        gradiva = naključna_neobiskana(u_id, r_id)
25:      shrani_način_priporočanja_in_priporočilo()
26:      # algoritem vrne samo seznam enoličnih identifikatorjev gradiv
27:      return gradiva

```

uporabnik ne more vplivati na ocenjevanje gradiv, razen če namenoma reši nalogo narobe. Mogoče bi bila bolj smiselna tristopenjska lestvica v smislu neustrezno gradivo, nevtralnno mnenje in dobro gradivo, a v tem primeru bi bila subjektivnost verjetno še bolj prisotna. Veliko uporabnikom se namreč gradivo mogoče ne bi zdelo dovolj dobro, saj bi radi imeli več animacij ali kaj podobnega. Vsi omenjeni razlogi so vplivali na odločitev, da smo izbrali zgolj oceni 0 in 1, ki pa sta uporabniku povsem skriti.

Matrika je zgrajena iz vseh uporabnikov in vseh gradiv. Posamezna vrstica tako predstavlja uporabnika, stolpec pa gradivo. Matrike so tipično zelo redke, ker imamo po navadi za vsakega uporabnika malo ocen. V našem primeru matrika ni tako redka, ker imamo samo 26 gradiv in ker je uporabnik v povprečju predelal malo manj kot 14 gradiv. To pomeni, da imamo za vsakega uporabnika v matriki približno polovico vrstice zapolnjene z vrednostmi. V primeru, da bi v sistem dodali še ogromno gradiv, bi bila matrika bolj redka. Pri matrični faktorizaciji je bila uporabljena metoda inkrementalne hkratne matrične faktorizacije (ISMF), kjer inicializiramo matriki P in Q , ki ju potem v vsaki iteraciji popravljamo tako, da bo napaka manjša [28].

Vsakič, ko se je generirala nova napoved, je bila uporabljena cela množica, dopolnjena z novimi podatki, ki smo jih dobili v vmesnem času od prejšnje napovedi. To je bilo mogoče, ker nismo imeli tako velikega števila uporabnikov in gradiv. V primeru, da bi bila matrika bistveno večja, bi lahko prišlo do prevelike zakasnitve, uporabniki pa bi morali čakati, preden bi se stran z novimi priporočili odprla. V takem primeru bi morali napovedi generirati samo na vsake toliko časa – to bi eksperimentalno določili.

Ob vsaki napovedi smo izračunali napoved za vsa gradiva in jih potem razvrstili od najpomembnejše do najmanj ustrezne. Napovedi za gradiva, ki jih je dijak že pogledal, smo izločili.

Priporoči gradiva v primeru nepravilne naloge – ta način priporočanja se uporabi, kadar uporabnik ne odgovori pravilno na zadnjo nalogo. Uporabniku želimo ponovno ponuditi gradivo, ki ima čim bolj podobno vsebino. V tem primeru se uporabi iskalna platforma Apache Solr,

ki omogoča vračanje podobnih dokumentov. To izvaja na način, da ima za vsak dokument pripravljen seznam besed in število njihovih pojavitev v posameznem dokumentu. Nato vrne dokumente, ki se medsebojno s seznamami besed najbolj ujemajo. Tak način delovanja našemu sistemu ustreza, zato smo ga tudi uporabili. Razne kratke besede (npr. vezniki, predlogi ...) so izločene tako, da ne vplivajo na podobnost dokumentov. Tako lahko učenec še nadoknadi, kar je pri prejšnjem gradivu slabše predelal, ali pa se s tem pokaže, da določenega sklopa gradiv enostavno ne obvlada. Slabost načina je, da trenutna gradiva v našem sistemu sicer so lahko iz istega sklopa, ni pa nujno, da ima drugo gradivo ravno tisti delček vsebine, ki ga je pokrivala naloga pri prejšnjem gradivu. Poleg tega ima sistem še eno slabost, in sicer da nimamo več podobnih gradiv, ker je lahko učenec že vse pregledal, ali pa da za obstoječega nimamo nobenega dovolj podobnega. To se je zgodilo samo zaradi tega, ker smo imeli zelo majhno število gradiv.

Najbolj obiskana gradiva – ta način se uporabi, kadar nimamo še nobenih podatkov o podanem uporabniku. To pomeni, da uporabnik ni pogledal še nobenega gradiva v našem sistemu. V tem primeru vrnemo gradiva, ki so bila s strani ostalih uporabnikov največkrat obiskana. V primeru, da nimamo niti podatkov o drugih uporabnikih, kar se je zgodilo na začetku uporabe našega sistema, izvede naključni način priporočanja.

Naključna gradiva, še neobiskana – v tem načinu uporabniku vrnemo vsa gradiva, ki so v podanem razredu. Gradiva, ki jih je podani uporabnik že pregledal, odstranimo. Seznam enoličnih identifikatorjev gradiv na koncu razvrstimo v naključnem vrstnem redu.

Shranjevanje načina priporočanja in priporočil – na koncu si v bazo shranimo vsako priporočilo in podatek, kateri način priporočanja je bil izveden, tako da bomo lahko analizirali rezultate in uspešnost sistema.

4.1 Cilj testiranja

Primerjanje uspešnosti priporočanja bomo izvedli na podlagi za ta namen pogosto uporabljenih mer: $\text{preciznost}@n$ (formula 4.1), $\text{priklic}@n$ (formula 4.2) in ocena F1 (formula 4.3). Poleg teh mer smo izračunali še: povprečni rang (formula 4.4) in DCG (angl. discounted cumulative gain)(formula 4.5). Za vsako priporočilo posebej smo izračunali vse našteje mere in v rezultatih podali njihove povprečne vrednosti. Kot pravo obravnavamo samo tisto, na katero uporabnik klikne, gledamo pa prvih n , kjer je n parameter metrike.

$$\text{Preciznost}@n = \frac{\text{število pravih priporočil do } n - ja}{\text{število priporočil}} \quad (4.1)$$

$$\text{Priklic}@n = \frac{\text{število pravih priporočil do } n - ja}{\text{št. vseh pravih gradiv v sistemu}} \quad (4.2)$$

$$\text{Ocena F1}@n = \frac{2 \times \text{Preciznost}@n \times \text{Priklic}@n}{\text{Preciznost}@n + \text{Priklic}@n} \quad (4.3)$$

$$\text{Povprečni rang} = \frac{\sum_{i=1}^n i}{n} \quad (4.4)$$

$$\text{DCG} = \text{rel}_1 + \sum_{i=2}^n \frac{\text{rel}_i}{\log_2(i)}; \text{rel}_i \in 0, 1 \quad (4.5)$$

kjer izraz rel_i zavzame vrednost 0, če gre za nerelevanten predlog, in 1, če gre za relevanten predlog. Kot pravo priporočilo smo obravnavali tisto, na katero je uporabnik kliknil.

Prvi cilj testiranja je bil ugotoviti, kateri način priporočanja je najboljši in ali je smiselno, da bi se potem dejansko uporabljal. Drugi cilj je bil primerjava naših rezultatov z ostalimi deli. Tretji cilj je bil preverjanje in ovrednotenje uporabnosti učne analitike v sistemu. Zadnji, četrti, cilj, ki smo si ga zadali, je bil preverjanje negotovosti učencev pri odgovorih.

4.2 Protokol testiranja

Najprej smo eno skupino uporabili za testiranje, da smo določili, koliko časa potrebujejo za uporabo sistema in če vse deluje, kot mora. Omenjena skupina ni prišteta k podatkom o številu skupin in številu dijakov, ki so omenjeni v nadaljevanju (tabela 4.1). Prvi skupini smo dali na voljo 25 minut za reševanje, a izkazalo se je, da je to preveč časa, ker so medtem pregledali skoraj že večino gradiv. Naš cilj je bil, da jih omejimo na toliko časa, da bodo obdelali okvirno polovico gradiv. Če uporabnik namreč pregleda vsa gradiva vnaprej, priporočanje nima več smisla, saj je potem razlika samo v vrstnem redu obdelanih gradiv.

Za časovno omejitev smo se odločili zaradi tega, ker če bi rekli, naj vsak predela 10 gradiv, bi morali dijaki sami šteti, koliko so jih že obdelali, ali pa bi morali modificirati sistem. Določeni dijaki rešujejo hitreje kot ostali in bodo predelali več gradiv. Gradiva se medseboj razlikujejo tudi po svojem obsegu, zato smo določili okvirni čas, ki ga imajo na voljo za reševanje. Na podlagi prve (testne) skupine smo za vse ostale skupine postavili okvirni čas 15 minut, kar je lahko pomenilo tudi kašno minuto več ali manj.

Vse ostale skupine (6 skupin prvih letnikov in skupina četrth letnikov) so v 15 minutah testirale sistem. Vseh 7 skupin je testiralo sistem na enak način, na voljo pa so imeli ista gradiva. Vsak dijak je najprej dobil listek s svojim uporabniškim imenom in geslom, s katerima se je prijavil v sistem. Po opravljeni prijavi so morali odpreti še razred z imenom informatika. Prikazala so se jim gradiva, razvrščena v vrstnem redu glede na to, kar je sistem priporočil za vsakega uporabnika posebej. Dijaki so dobili priporočila glede na svoj uporabniški enolični identifikator po modulu 3, kar lahko podrobneje razberemo iz algoritma 1. Cilji vseh skupin prvih letnikov so bili enaki, in sicer da uporabljajo sistem, upoštevajo priporočena gradiva in kliknejo na gradivo, ki je zanje v tistem trenutku najbolj primerno, tudi če se je mogoče sistem zmotil. Cilj skupine iz četrth letnikov je bil ponavljanje snovi za maturo, saj so vso snov že predelali, samo določene stvari so že malo pozabili.

Vsi dijaki so imeli uporabniška imena, tako da je vse potekalo anonimno.

4.3 Pričakovanja

Najbolj nas je skrbelo, če bodo dijaki dovolj resno uporabljali sistem, saj bi v nasprotnem primeru dobili zelo čudne ali pa neuporabne rezultate. Ravno tako nas je skrbelo, da gradiv sploh ne bodo prebrali, temveč samo klikali naprej, da bi prišli do naloge. Pričakovali smo, da bodo bolj pregledovali gradiva s snovjo, ki je še niso obdelali, vendar se je izkazalo drugače.

Testni skupini smo dali na voljo več časa, ker smo pričakovali, da ga bodo toliko potrebovali, a izkazalo se je, da je 25 minut preveč. Mislili smo tudi, da so navodila, ki so jih dobili v prvi skupini, dovolj jasna, a očitno niso bila, ker so bolj ali manj klikali na prvo gradivo.

Glede na hitrost reševanja in omejen čas nismo pričakovali, da si bodo nekateri dijaki pri nalogi vzeli še čas in menjavali odgovore. Mislili smo, da bodo skoraj vsi odgovor zmeraj izbrali v prvo. Presenetilo nas je tudi, da so naloge precej uspešno reševali.

Pričakovali smo bistveno večjo preciznost in boljše rezultate v primerjavi z ostalimi deli glede na to, da so v nekaterih delih uporabljali zelo preproste načine priporočanja. Matrična faktorizacija pa bi morala dati boljše rezultate v primerjavi z bolj preprostimi načini priporočanja.

4.4 Potek poskusov

Testiranje je potekalo od 30. 1. 2017 do 3. 2. 2017 v 6 skupinah prvih letnikov in skupini četrtega letnika. Skupaj je tako sodelovalo 85 dijakov. Testna skupina ni prišteta k omenjenim skupinam in številu dijakov.

Za hitrejšo izvedbo testiranja smo uporabniška imena in naključno generirana gesla že vnaprej pripravili, jih natisnili in na listkih razdelili dijakov. Uporabniška imena so bila sestavljena v obliki student13@student13.si, kjer so se spreminjale zgolj številke. Na ta način je bilo zagotovljeno, da so zadeve potekale anonimno, in hkrati tudi, da dijaki niso uporabljali istega uporabniškega imena. S tem, ko smo vnaprej pripravili uporabniške račune, smo se izognili nepotrebnim izgubi časa, ki bi ga dijaki porabili za ustvarjanje

uporabniškega računa v sistemu.

Dijaki so dobili navodila, kako morajo sistem uporabljati, in sicer da jim sistem predlaga gradiva, za katera misli, da so zanje v tistem trenutku pomembna. Na začetku so gradiva, za katera sistem misli, da so bolj primerna, vsako naslednje gradivo pa je malo manj primerno. Sistem vam lahko priporoči tudi neoptimalno gradivo, zaradi česar so morali izbrati ustrezno gradivo. Dijaki so imeli na voljo 15 minut časa za uporabo sistema.

Po hitrem pregledu rezultatov prve (testne) skupine se je izkazalo, da smo pripravili nejasna navodila, ker so dijaki večinoma zmeraj kliknili na prvo gradivo. Sicer je res, da je gradivo na prvem mestu tisto, ki ga sistem priporoča kot najboljšega, a na koncu ne bi dobili kakšnih uporabnih podatkov. Zaradi tega smo pripravili jasnejša navodila o tem, kako sistem deluje in da morajo vsakič pregledati gradiva, ki jih priporoča, in izbrati tisto, ki je dejansko primerno, tudi če ni na prvem mestu. Zaradi navedenih razlogov in drugačne časovne omejitve podatkov testne skupine nismo uporabili pri rezultatih.

V posamezni skupini je bilo do 15 dijakov – število je bilo odvisno od tega, koliko jih je bilo tisto uro prisotnih pri pouku, in od oddelka, saj jih v nekaterih oddelkih ni 15. V vseh skupinah je bilo izvedeno priporočanje na enak način, ki je bil odvisen od uporabnikovega enoličnega identifikatorja, kot je to razloženo v algoritmu 1.

4.5 Rezultati

Vse skupine je učil isti profesor, tako da so predelali isto snov in imeli enako znanje. Glede znanja se razlikuje edino skupina četrtil letnikov, ki je vso snov predelala že v prvem letniku.

V nadaljevanju opisujemo podatke iz tabele (tabela 4.2) o številu priporočil glede na tip priporočanja. Podobno gradivo je bilo uporabljeno samo, kadar je dijak rešil nalogo nepravilno. Poleg tega je zaradi majhnega števila gradiv prišlo do tega, da dovolj podobno gradivo ni bilo zmeraj na voljo,

Podatek	Vrednost	Opis
Število dijakov	85	Število dijakov, ki je bilo udeleženo v testiranju.
Število gradiv	26	Število gradiv, ki so jih dijaki imeli na voljo.
Čas	15 min.	Čas, ki so ga dijaki imeli na voljo za uporabo sistema.
Število obdelanih gradiv	1171	Skupno število vseh obdelanih gradiv.
Povprečno število pregledanih gradiv	13,78	Število gradiv, ki jih je v povprečju pogledal posamezen dijak.
Povprečen čas, ki so ga porabili za posamezno gradivo	39,87 s	Povprečen čas, ki so ga dijaki porabili za posamezno gradivo.
Število pravilno rešenih nalog	981	Skupno število nalog, ki so bile rešene pravilno.
Število napačno rešenih nalog	225	Skupno število nalog, ki niso bile rešene pravilno.
Odstotek pravilno rešenih nalog	81,34 %	
Število vseh odgovorov	1206	Število vseh odgovorov – rešenih nalog.
Število menjav odgovorov	99	Skupno število vseh menjav odgovorov.
Odstotek menjav	8 %	Odstotek menjav odgovorov.
Število priporočil	1251	Skupno število vseh priporočil, ki jih je naredil sistem. Opomba število priporočil je večje, ker so to vsa priporočila, tudi tista, katerih uporabniki niso več uporabili, ker niso imeli več časa.

Tabela 4.1: Splošni podatki o testiranju

zaradi česar se ni uporabil ta način priporočanja. Največkrat obiskano gradivo je bilo uporabljeno samo prvič, ko o uporabniku nismo imeli še nobenih podatkov. Upoštevati moramo tudi, da v prvi skupini v sistemu še ni bilo največkrat obiskanih gradiv, zato za prvo skupino sistem ni uporabil tega načina priporočanja. Naslednji tip je naključno gradivo, če ni uspel noben drugi tip priporočanja. Ta način je bil uporabljen v prvi skupini, ko še nismo imeli na voljo podatkov o najbolj obiskanih gradivih in so dijaki dobili gradiva razvrščena v naključnem vrstnem redu. Poleg tega je bil način uporabljen tudi, če je uporabnik rešil nalogo nepravilno in sistem ni imel na voljo nobenega dovolj podobnega gradiva.

Zadnja tri priporočanja, ki so omenjena v tabeli (tabela 4.2) smo uporabili pri eksperimentu. Naključno gradivo se od prej omenjenega tipa naključnega gradiva (ki je bil v uporabi na začetku, ko nismo imeli še nobenih podatkov v sistemu) razlikuje v tem, da so pod to postavko zbrani uporabniki, pri katerih je bil v rabi naključni način. Kot vidimo, ima ta naključni način manjše število priporočil, kot jih imata naslednja dva. Na naključni način se je priporočalo 23 dijakom, medtem ko se je najbolj podobno gradivo priporočalo

Tip priporočanja	Št. priporočil	Št. priporočil, na katere je uporabnik kliknil	Opombe
Podobno gradivo glede na vsebino	60	50	
Največkrat obiskano gradivo	66	65	
Naključno gradivo	186	129	Naključno gradivo, če ni noben tip priporočanja uspel.
Naključno neobiskano gradivo	259	234	Naključno gradivo za uporabnike, ki jim je bilo priporočano na naključni način.
Najbolj podobno gradivo	343	307	
Najmanj podobno gradivo	337	288	

Tabela 4.2: Število priporočil glede na tip priporočanja in število priporočil na katere je uporabnik dejansko kliknil

28 dijakom in najmanj podobno 26 dijakom. Zastavili smo si sicer, da bi vsakega od treh načinov preizkušala tretjina dijakov. Ampak ker nismo strogo pazili pri razdeljevanju listkov, je prišlo do tega, da število uporabnikov ni bilo razdeljeno točno po tretjinah.

Največkrat obdelana so bila naslednja tri gradiva: vrste računalnikov, spletne nevarnosti in avtorske zaščite. Najmanjkrat je bilo predelano gradivo z naslovom *tiskalnik*. Najbolj obiskano gradivo je bilo predelano 77-krat, medtem ko je bilo najmanjkrat predelano gradivo predelano zgolj 27-krat. Število obdelav ostalih gradiv je v tem razponu. Iz podatkov lahko razberemo, da ni gradiva, ki bi ga dejansko predelali vsi dijaki. Posamezno gradivo je v povprečju predelalo 46,38 dijakov, kar je malo več kot polovica.

Ob pogledu na to, kaj se je dogajalo v povezavi z obsežnostjo gradiva (števila prosojnic), ugotovimo, da so bila bolj obsežna gradiva predelana redkeje kot gradiva, ki so bila krajša, npr. samo ena prosojnica. Najde se seveda tudi kakšna izjema. To se sklada z današnjim svetom, kjer gre vse na hitro, v katerem za natančno in poglobljeno delo ni časa in kjer je vse instantno. Hkrati kaže tudi na to, da mora biti vse na kratko razloženo, ker se nobenemu ne da več brati na dolgo. Obsežnost gradiva dijaku ni bila vidna in je lahko samo iz naslova sklepal na bolj obsežno gradivo.

Pri pregledu naslovov gradiv smo ugotovili, da so dijaki izbirali gradiva s

snovjo, ki so jo že obdelali in so jo že poznali. Vsebine, ki so bile povezane s strojno opremo, dijaki pri pouku še niso obdelali in zanimivo je, da je bila večina teh vsebin najmanjkrat predelana v primerjavi z ostalimi. Pri spremljanju testiranja smo opazili še, da so dijaki četrtil letnikov gradiva predelovali skoraj zaporedoma, torej so večinoma izbrali kar gradivo, ki ga je sistem predlagal na prvem mestu.

4.5.1 Negotovost dijakov

V sistemu smo beležili tudi negotovost učenca. Shranili smo vsak odgovor, ki ga je izbral, preden je oddal nalogo, in tako prešteli število menjav. Menjav odgovorov je bilo za 8 %, medtem ko so ostali izbrali odgovor v prvo. Kar 49 dijakov je vsaj enkrat zamenjalo odgovor, medtem ko so preostali učenci vedno izbrali odgovor v prvo in niso spreminjali odgovora. Od 49 je 18 dijakov odgovor zamenjalo več kot enkrat. Iz podatkov razberemo, da so tisti, ki so zamenjali odgovor, to naredili enkrat ali dvakrat. Več kot dveh menjav je bilo malo.

Če primerjamo število menjav pri posameznem dijaku s številom gradiv, ki jih je predelal, ugotovimo, da dijak zamenja vprašanja največ pri tretjini gradiv. Največ menjav odgovorov je bilo pri gradivu vrste informacij in zvezno, diskretno – pri obeh je odgovore menjavalo 7 dijakov. Vprašanja pri teh dveh nalogah po našem mnenju sicer nista bili najtežji, če ju primerjamo z ostalimi nalogami. Poleg tega so snov, ki je pokrivala vprašanja, že prej predelali.

Negotovosti uporabnikov sicer nikjer v sistemu ne prikazujemo, tudi učitelju ne. Lahko pa bi to prikazali kot dodatno informacijo v okviru učne analitike.

Določene naloge je več dijakov rešilo pravilno kot ostale, ampak odstopanje spet ni tako bistveno. Edino naloga pri gradivu sistemska programska oprema zelo izstopa, ker je bilo vprašanje malo težje.

4.5.2 Uspešnost priporočanja

Preciznost za posamezno priporočilo smo izračunali na podlagi števila gradiv, vključno s tistim, na katerega je dijak kliknil. To smo potem delili s številom vseh gradiv, ki so bila dijaku priporočena.

Način priporočanja	Preciznost@1	Priklic@1	Ocena F1@1
Naključno neobiskano gradivo	0,3166	0,3388	0,3273
Najbolj podobno gradivo	0,0292	0,0413	0,0342
Najmanj podobno gradivo	0,1573	0,2190	0,1831

Tabela 4.3: Tabela povprečnih preciznosti za vsak način priporočanja, če je $n = 1$

Način priporočanja	Preciznost@5	Priklic@5	Ocena F1@5
Naključno neobiskano gradivo	0,4595	0,2500	0,3238
Najbolj podobno gradivo	0,1808	0,1305	0,1514
Najmanj podobno gradivo	0,4273	0,3025	0,3542

Tabela 4.4: Tabela povprečnih preciznosti za vsak način priporočanja, če je $n = 5$

V naših poskusih oceni F1@1 in F1@10 kažeta, da je najboljši način priporočanja naključni, takoj za njim pa najmanj podobno gradivo, ki priporoča čim bolj drugačno gradivo (tabela 4.3, 4.4, 4.5, 4.6, 4.7). Ocena F1@5 prikazuje drugačen rezultat, in sicer, da je najboljše priporočanje najmanj podobnega gradiva. Sledita mu še naključni način priporočanja in priporočanje najbolj podobnega gradiva. Matrična faktorizacija, ki priporoča čim bolj podobno gradivo, je po naših rezultatih na zadnjem mestu. Iz tega sledi, da

Način priporočanja	Preciznost@10	Priklic@10	Ocena F1@10
Naključno neobiskano gradivo	0,6602	0,2548	0,3677
Najbolj podobno gradivo	0,3907	0,1997	0,2643
Najmanj podobno gradivo	0,5430	0,2727	0,3631

Tabela 4.5: Tabela povprečnih preciznosti za vsak način priporočanja, če je $n = 10$

Način priporočanja	Povprečni rang
Naključno neobiskano gradivo	6,7051
Najbolj podobno gradivo	11,8632
Najmanj podobno gradivo	8,2500

Tabela 4.6: Tabela povprečnih rangov

moramo dijaku ponuditi naključno gradivo. Pričakovali smo sicer, da bo čim bolj podobno gradivo vseeno pokazalo precej boljši rezultat kot naključni način, in ravno tako, da ne bosta najboljša dva načina po rezultatu tako tesno skupaj.

Za primer napačno rešene naloge se je uporabniku priporočalo podobno gradivo. Ta način je vrnil samo eno ali dve gradivi, kar je posledica majhnega števila gradiv v našem sistemu. Ker so v primeru, ko je sistem ponudil dve gradivi, uporabniki vedno kliknili na prvo, preciznosti nismo računali. Na začetku, ko o posameznem uporabniku še nismo imeli podatkov, se je uporabil način, ki je vrnil največkrat obiskana gradiva. Preciznost in priklic za $n = 10$ sta 0,3333 in 0,0298, torej precej nižja od matrične faktorizacije. Omeniti moramo še to, da je bil ta način pri uporabniku uporabljen samo prvič. Tako je bilo teh napovedi precej manj v primerjavi z ostalimi načini. Drugi razlog je, da je bilo veliko klikov na prvo gradivo, ker pač prvič enostavno klikneš na prvo, ker še ne veš, kako sistem deluje.

Način priporočanja	DCG
Naključno neobiskano gradivo	0,5484
Najbolj podobno gradivo	0,3250
Najmanj podobno gradivo	0,4792

Tabela 4.7: Tabela rezultatov mere DCG

Preciznost je sicer na splošno zelo nizka. To pripisujemo predvsem dejstvu, da je bilo v sistemu malo gradiv. Če bi jih imeli precej več, bi bila verjetno večja. Na izbor gradiv sta lahko vplivala tudi slika pri posameznem gradivu in naslov posameznega gradiva. Vpliva lahko tudi to, koliko gradiv je prikazanih v prvi vrstici, pa tudi to, da tipično gledamo zmeraj zgornji levi del zaslona. S tega stališča je priporočilo, ki je bilo v drugi vrstici na prvem mestu, bolj pomembno kot priporočilo, ki je bilo na koncu prve vrstice in v desnem delu zaslona. A tega pač nismo mogli izmeriti, saj bi morali s kamero spremljati pogled po zaslonu.

4.5.3 Uporabnost učne analitike

Uporabnosti učne analitike ne moremo izmeriti, temveč jo lahko samo komentiramo. Težava pri učni analitiki je bila, da smo imeli nastavljeno prikazovanje podatkov o napredku v zadnjih 24 urah. V dnevu so sistem testirale tri skupine, drugi dan pa recimo samo 1. V tem primeru ali pa, ko je bila več kot ena skupina, se rezultati med obema skupinama ali skupinami v okviru učne analitike niso ločili. To bi sicer lahko enostavno rešili tako, da bi učno analitiko prikazovali samo za zadnjo uro. Ker pa smo podatke prikazovali za zadnjih 24 ur, je bilo mogoče že malo preveč podatkov, zaradi česar se je tudi čas nalaganja povečal in ni bilo več tako sprotno.

Druga težava pri prikazu je bila posledica tega, da gradiva niso bila zelo obsežna in so jih dijaki veliko pregledali, zaradi česar je bilo prikazanih veliko podatkov o obdelanih gradivih. Ugotovili smo tudi, da bi bilo boljše, če bi v tem pogledu odstranili meni in naredili prostor za nadzorno ploščo. Tako bi bili grafi takoj vidni in ne bi bilo treba iti z drsnikom nižje. Razporeditev

bi torej morali malo spremeniti. Ker je bilo testiranje sistema anonimno, v učni analitiki ni bilo videti, za katerega dijaka gre.

Pri učni analitiki bi lahko prikazali tudi druge podatke, recimo v povezavi z ocenami. A to je spet težava, saj bi morali imeti vse podatke v sistemu. Zato takih prikazov tudi nismo delali in smo se omejili na podatke, ki smo jih zbrali v sistemu.

4.6 Pomen rezultatov

Izkazalo se je, da sta najboljša načina priporočanja naključno gradivo in čim bolj drugačno gradivo. Zanimivo je, da je naključni način priporočanja zelo blizu načinu priporočanja čim bolj drugačnega gradiva.

Upoštevati moramo, da smo imeli zelo majhno število gradiv v primerjavi z drugimi deli in da se gradiva niso povezovala med seboj. Če bi za posamezno gradivo veljalo, da je treba predelati katero drugo gradivo, bi morali spremeniti sistem in ravno tako priporočanje.

Pri napačno rešeni nalogi bi bilo boljše, če bi uporabili enega od treh načinov priporočanja in samo na prvo mesto postavili podobna gradiva. V nadaljevanju bi bila prikazana gradiva glede na način, ki se izbranemu uporabniku priporoča. Do te ideje smo prišli že po testiranju in pregledu rezultatov.

Nato, da se je kot najboljši način priporočanja izkazal ravno naključni, ne moremo biti ponosni, ker je to zelo slabo. Pričakovanja so bila, da bo namreč matrična faktorizacija, ki vrača čimbolj podobno ali pa čim bolj drugačno gradivo, izkazala kot najboljši način. Sam naključni način smo vključili samo zaradi primerjave glede na ostale načine, ampak na koncu se je žal izkazal kot najboljši. Ena od možnosti bi seveda bila, da bi namesto naključnega načina uporabili kakšnega drugega. Vsekakor bi morali testirati še več načinov in ugotoviti, zaradi česa je prišlo do tega. Ravno tako bi morali ugotoviti, kaj bi lahko naredili z našo učno množico, da ne bi prišlo do takšnih rezultatov. Poleg tega bi morali poiskati še več del, ki se ukvarjajo

s priporočanjem in ne samo s priporočanjem na področju učnih sistemov.

Priznati moramo tudi, da se matrična faktorizacija pogosteje uporablja z bistveno večjim naborom podatkov, ki že po sami količini uporabnikov bistveno presega število vseh učencev pri predmetu *informatika* v Sloveniji. Morda bi se morali bolj osredotočiti na algoritme, ki daje boljše rezultate na majhnih naborih podatkov.

Možno je tudi, da bi dobili drugačne rezultate, če pri samih gradivih ne bi imeli slik, ker to sigurno tudi vpliva na izbiro gradiva. Mogoče bi bilo dobro, da bi priporočili samo nekaj gradiv takoj po sami nalogi, če pa učenec ne bi bil z nobenim priporočilom zadovoljen, bi pa mu prikazali vsa gradiva. Ravno tako je lahko bilo za slab rezultat krivo to, da so določene vsebine že poznali in so jih z uporabo našega sistema zgolj ponovili.

Pri prikazu učne analitike v sistemu smo naredili to napako, da nismo prikazali vse v velikosti zaslona. Grafi so se nahajali nižje in je bilo potrebno uporabiti drsnik, da smo prišli do določenih informacij. Ravno bi morali učno analitiko prikazovati samo za zadnjo uro, ker ko sta sistem testirali dve skupini v enem dnevu, so bili tako v prikazu vključeni še rezultati prejšnje skupine. V učno analitiko bi lahko vključili tudi negotovost učencev glede na to, da smo te podatke zbirali.

4.7 Primerjava z drugimi deli

V različnih delih so svojo uspešnost merili na različne načine. Zato se po vrednostih zelo težko primerjamo z drugimi deli. Opisali bomo le ugotovitve, ki so pomembne v primerjavi z drugimi deli.

4.7.1 Primerjava priporočilnega sistema

V delu [15] so uporabnike v dveh vprašanjih spraševali, če so naloge razumeli. Mi smo razumevanje naloge preverjali na naslednji način: če so nalogo pravilno rešili, smo to šteli, kot da so jo razumeli, v nasprotnem primeru pa ne. S tem smo se želeli izogniti anketam. Ravno tako si nismo želeli pristran-

skosti uporabnika, ki je vedno prisotna z raznimi vprašanji. V omenjenem delu so spreminjali tudi težavnost gradiva na podlagi vprašanj in jo prilagajali glede na zmožnost uporabnika. V našem delu gradiva nimajo različnih težavnosti, tako da se glede težavnosti in tudi glede na zmožnost uporabnika ne moremo primerjati. Za potrebe testiranja so imeli samo 9 gradiv več, kot jih imamo mi, in skoraj trikrat več uporabnikov, ki so testirali njihov sistem. Naš odstotek (81,34 %) pravilno rešenih nalog je zelo blizu odstotku (82,50 %) razumevanja nalog v omenjenem delu.

Podoben povprečen odstotek (79,60 %) končnih preverjanj so dobili tudi v delu [17]. Za razliko od našega so morali uporabniki v delu [17] narediti še predtest, na podlagi katerega so zbrali podatke. V zadnjem omenjenem delu so se bolj ukvarjali s preverjanji znanja, ne pa toliko z gradivi, na podlagi katerih bi preverili znanje.

V delu [16] so uporabljali pet različnih načinov za priporočanje. Njihovo uspešnost so ocenjevali s preciznostjo in oceno F1. Njihov nabor podatkov je bil v primerjavi z našim zelo obsežen, saj je obsegal skoraj 700 uporabnikov in malo manj kot 3800 knjig. Z delom se lahko primerjamo v tem, da smo uporabili več načinov priporočanja. Numerično se lahko primerjamo s primerjavo vrednosti preciznosti in ocene F1. V njihovem sistemu se je kot najboljši način priporočanja izkazala kombinacija najbolj podobnih obiskanih gradiv in najbolj podobnih uporabnikov (preciznost 0,7540). Priporočanje podobnih stvari se je izkazalo kot najslabši (preciznost 0,5840) način priporočanja. Isto velja tudi za oceno F1, le številke so drugačne.

V našem delu smo dosegli bistveno slabše rezultate. Naš najboljši rezultat glede na preciznost je 0,6602, kar je slabše. Glede na to, da v primerjanem delu niso uporabljali nobene matrične faktorizacije, temveč samo različna priporočanja na podlagi atributov in stvari, so naši rezultati pre-slabi. V splošnem bi morali z načini matrične faktorizacije dobiti bistveno boljše rezultate, kot smo jih dobili na podlagi podobnosti atributov in stvari na podlagi različnih mer. Preciznost bi bila ob drugačnem načinu prikaza gradiv, z več uporabniki in gradivi v našem sistemu verjetno višja. Čeprav v

primerjanem delu niso uporabljali matrične faktorizacije, naša preciznost ni dosti višja od pričakovane.

V drugem delu istega avtorja [6] so priporočali glede na podatke o poti (zaporedju med gradivi). Na podlagi zaporedij so zgradili drevo, s pomočjo katerega so novim uporabnikom glede na njihovo trenutno pot napovedali nova gradiva. Tudi njihov nabor podatkov je bil zelo obsežen, saj je vključeval skoraj 2000 uporabnikov in skoraj 3000 objektov. Ugotovili so, da je najboljši hibridni način med zaporedji med gradivi in atributi stvari. Dosegli so preciznost 0,7520. V delu je narejena tudi primerjava glede na preciznost in oceno F1 z različnimi načini priporočanja. V primerjavi z našim delom vsi omenjeni načini dosegajo bistveno višjo preciznost. Njihov najslabši način priporočanja glede na zaporedje med gradivi je dosegel preciznost 0,5470. V primerjavi z njimi smo imeli zelo majhen nabor podatkov in se nismo ukvarjali s priporočanjem glede na pot.

Z delom [23] se prav tako lahko primerjamo v preciznosti. Ugotovili so, da je bila preciznost precej večja v primeru, ko so iskali podobna gradiva v kombinaciji z oceno uporabnikov. Izbiranje samo podobnih gradiv da bistveno slabše rezultate. Izračun preciznosti v njihovem delu se razlikuje od našega, ker so kot prava priporočila šteli vsa, ki so imela oceno več kot 0,7. V delu opozarjajo na pristranskost uporabnika ob ocenjevanju gradiv, ki smo se ji izognili tako, da uporabniki niso neposredno ocenjevali gradiv. V primerjavi z našim sistemom so imeli v svojem približno petkrat več gradiv in malo manj uporabnikov.

Učno analitiko smo implementirali malo drugače kot v drugih delih. V našem e-učnem sistemu je na voljo samo za učitelja. Ravno tako nismo iskali povezav med učenci, kot so to delali v delu [14]. V našem prikazu tudi ni enostavno vidno, koliko gradiv je učenec pregledal.

Primerjamo se lahko še z delom [22], kjer so učne sloge posameznika identificirali avtomatsko in na podlagi vprašalnika. Da so sploh lahko identificirali različne učne sloge, je moral sistem obsegati več funkcionalnosti (npr.: forum), kot jih obsega naš. Njihov sistem je uporabljalo 127 študentov, kar

je malo več kot pri nas. Žal ni podatka o številu gradiv. Preciznost so sicer računali malo drugače, a njihov najboljši rezultat je 0,7933.

Podobno so tudi v delu [21] identificirali učne sloge in priporočali glede na zaporedje obiskanih gradiv. Njihov sistem je testiralo petkrat več uporabnikov kot našega in na voljo so imeli veliko več gradiv. Za razliko od prej omenjenih del za ocenjevanje uspešnosti niso uporabljali preciznosti.

V delu [24] so zbirali podatke iz dnevniških datotek, da so dobili tudi pot uporabnika. Priporočanje so izvedli na hibridni način, in sicer v kombinaciji med skupinskim izbiranjem in na podlagi vsebine. Njihov nabor podatkov je bil precej večji od našega, saj je vseboval podatke o okoli 12.000 sejah. Uporabljali so različno široka okna, s pomočjo katerih so zajemali zaporedja gradiv, in potem priporočali na hibridni način. Ugotovili so, da s širšim oknom dobijo večjo preciznost. Z delom se lahko primerjamo v preciznosti; dosegli so preciznost med 0,2 in 0,4. Podatke o preciznosti navajajo samo na grafu, tako da lahko razberemo, da so naši rezultati v preciznosti podobni kot pri njih oziroma v določenih načinih boljši.

Naši rezultati so še boljši kot rezultati v delu [29], kjer sicer ne gre več za priporočanje v sistemu za učenje, temveč za priporočanje v digitalni knjižnici. Najboljši način priporočanja v delu je bil glede na podatke o uporabnikih (preciznost 0,33). Njihov najboljši način je še vedno malo slabši od našega. Poleg tega načina so priporočali še glede na metapodatke in celoten nabor podatkov. Tudi pri njih gre za bistveno obsežnejši nabor podatkov, kot smo ga imeli mi.

Tudi v delu [30] so uporabljali hibridni način priporočanja, in sicer v kombinaciji med pravili in na podlagi vsebine. Tukaj sicer ne gre za enak način priporočanja kot v ostalih delih, temveč za semantično iskanje gradiv, tako da uporabnik dobi pravo gradivo. Največja preciznost, ki so jo dosegli, je 0,25, kar je slabše v primerjavi v našim delom.

Malo bolj drugačen način priporočanja glede na ostala dela so uporabili v delu [31], kjer so priporočali na podlagi trenutne strukture predmeta in tipa učnega gradiva. To so vključili v formule, na podlagi katerih so dobili

priporočilo. Nabor podatkov je vključeval 18 enot, ki so imele skupaj 300 gradiv. Preciznost glede na testiranje različnih načinov se giblje med 0,2704 in 0,5926, kar pomeni, da smo v določenih primerih boljši.

Omenili smo že delo, kjer so priporočali na podlagi metapodatkov. Enako so naredili tudi v delu [32], in sicer na podlagi metapodatkov o gradivu, ki so jih lahko dobili na podlagi uporabljenega standarda LOM. Omenjeni standard je namenjen dodajanju metapodatkov gradivom v e-učnih sistemih. V eksperimentu, kjer je bil uporabljen standard LOM, so dosegli preciznost 0,9724. To je najboljši rezultat v preciznosti glede na vsa opisana dela. Mi dosežemo bistveno slabši rezultat, poleg tega pa naša gradiva niso imela dodanih metapodatkov.

Učne sloge so poskušali identificirati še v delu [33], in sicer s pomočjo nevronske mreže. Njihova najboljša preciznost znaša 0,8070. Sistem je testiralo 75 študentov, kar je malo manj kot pri nas. Za tako identifikacijo učnih slogov ima naš sistem premalo funkcionalnosti.

Na podlagi primerjav z vsemi deli smo ugotovili, da nam je skupno to, da je bilo tako v večini del kot pri nas preizkušenih več načinov priporočanja. Z večino del smo se lahko primerjali v preciznosti, a smo ugotovili, da smo v primerjavi z večino bistveno slabši. Glede nabora podatkov je tako, da ima nekaj del primerljivo število podatkov, ostala pa bistveno več. Predvsem imamo majhno število gradiv.

4.7.2 Primerjava učne analitike

Pri učni analitiki nimamo nobenih številskih rezultatov, s katerimi bi se primerjali z ostalimi deli, tako kot smo se prej glede na uspešnost priporočanja in nabora podatkov. Zato bomo primerjavo naredili samo opisno: kaj so naredili v drugih delih in kaj smo naredili mi.

V okviru učne analitike so v delu [12] prikazovali podatke o tem, koliko študentov je dostopalo do posameznega gradiva, graf ocen posameznih preverjanj, število aktivnih študentov v posameznem dnevu, število objav na forumu, uporabo posamezne funkcionalnosti sistema in najbolj obiskana

gradiva. V tem sistemu je mogoče dobiti še bolj podrobne prikaze glede na podane parametre (npr.: časovno obdobje ...). V našem sistemu prikazujemo v malo drugačni obliki število aktivnih študentov, uspešnost reševanja nalog in obiskana gradiva. Omenjeni sistem tako nima nekega sprotne pregleda nad stanjem v razredu in napredkom učencev, kar pa naš sistem ima.

V delu [13] je prikazana tudi učna analitika za učenca, ki prikazuje njegov napredek. V našem sistemu učenec nima na voljo nobene učne analitike, lahko vidi le rezultate, ki jih je dosegel pri nalogah. V omenjenem delu je prikazan še primer učiteljevega pogleda, v katerem je mogoče videti napredek razreda, aktivnost v njem in koliko stvari so že naredili. V primerjavi z našim delom ni vidno, koliko časa povprečno porabijo za posamezno dejavnost. V primerjavi z nami imajo boljši prikaz na nadzorni plošči; v našem sistemu se moramo z drsnikom pomakniti nižje, da vidimo še ostale grafe. V primerjanem delu vidimo še prikaz za posameznega učenca, in sicer njegovo uspešnost glede na posamezne parametre, česar v našem delu nimamo.

Zelo zanimiv pregled nad razredom so naredili v delu [14], kjer na grafu prikazujejo, kako napredujejo posamezni učenci. Označeno je tudi povprečje razreda, tako da je mogoče videti, če učenec zaostaja ali napreduje v primerjavi z ostalimi. Poleg tega prikazujejo še izoliranost posameznih učencev. V našem sistemu sicer prikazujemo napredek učencev, vendar pa iz prikaza ni neposredno vidno, kje je učenec v primerjavi z ostalimi. Ravno tako ne ugotavljamo izoliranosti oziroma sodelovanja učencev med seboj.

Primerjava glede učne analitike nam pove, da vsak prikaže stvari malo drugače. Vsepovsod ni mogoče v realnem času spremljati napredka študenta oziroma celega razreda. V nekaterih delih tudi ne moremo priti do določenih prikazov, ker pač niso na voljo in smo omejeni s tem, kar sistem ima.

V našem delu smo dosegli zelo slab rezultat glede načina priporočanja, ker se je namreč naključni način priporočanja izkazal kot najboljši. Zato, da učencem prikažemo gradiva v naključnem vrstnem redu pa ne rabimo nobenega priporočanja. Pričakovano je bilo, da bo matrična faktorizacija dala bistveno boljši rezultat kot naključni način priporočanja.

Poglavje 5

Sklepne ugotovitve

Različni e-učni sistemi bodo v šolah v prihodnosti vse bolj v rabi. Ravno tako se bodo še bolj prilagajali potrebam posameznega učenca glede na to, kaj že zna in kaj se mora še naučiti. Vsebine in gradiva bodo predstavljali v učnem slogu, ki je za posameznika najbolj primeren.

Priznati moramo, da naš sistem poruši tipičen vrstni red obravnavane snovi. Zaradi tega marsikdo nad tem že v osnovi ne bi bil navdušen. Še večja težava nastopi, če se snov zelo navezuje – za take primere bi morali sistem vsekakor prilagoditi. Z našimi gradivi težav načeloma nismo imeli, ker so bila že tako zasnovana, da niso odvisna drugo od drugega.

Naredili smo preprost e-učni sistem, ki smo ga tudi testirali. Za potrebe testiranja smo naredili gradiva, ki so pokrivala predmet informatika. Testiranje različnih načinov priporočanja je pokazalo, da sta najboljša naključno gradivo in čim bolj drugačno gradivo glede na ostale uporabnike, medtem ko se čim bolj podobno gradivo izkaže kot slabo. Iz tega izhaja, da učencev ne zanimajo že znane ali čimbolj drugačne vsebine, gradiva oziroma snov, temveč jim moramo priporočiti naključno gradivo.

V sistem smo implementirali sprotno učno analitiko, ki omogoča pregled napredka učencev. S tem lahko učitelj sproti zaznava, katere naloge slabo rešujejo, in učence tudi sproti opozori na določene stvari ali pa jim kakšno stvar dodatno pojasni. Ob koncu ure lahko naredi analizo in jim morebiti še

dodatno razloži stvari, ki so jim na splošno delale težave.

Naredili smo primerjavo z drugimi deli. Vsak je delal malo drugače in uporabljal drugačne načine priporočanja. Razlike so bile v obsegu sistema, številu uporabnikov in gradiv. Kjer smo se lahko primerjali z drugimi deli, smo to tudi opisali. Ugotovili smo, da v preciznosti dosežemo bistveno slabše rezultate v primerjavi z drugimi deli. To bi se verjetno delno izboljšalo, če bi imeli precej več gradiv in uporabnikov, a še vedno ne bi dosegli rezultatov, kot so jih drugi. V primerjavi z nekaterimi deli smo popolnoma skrili načine ocenjevanja gradiv in njihovega razumevanja. Na ta način smo se znebili pristranskosti uporabnika, saj mu gradiv ni bilo treba ocenjevati na lestvici, temveč je to naredil na prikrit način s tem, ko je rešil nalogo. Prav tako nismo izvajali nobenih anket o uporabi sistema, kot so to delali v nekaterih drugih delih. Uporabniki tudi niso vedeli, da jim sistem priporoča gradiva na različne načine.

Vsekakor je težava dobiti dovolj testnih uporabnikov za uporabo sistema, kar pa velja tudi za druge podobne sisteme. Z večjo količino uporabnikov bi vsekakor dosegli tudi boljšo točnost. Za izboljšavo same točnosti bi bilo smiselno testirati več algoritmov, ki uporabljajo matrično faktorizacijo in delujejo dobro tudi z majhnim naborom podatkov.

Naš sistem je zasnovan tako, da bi načeloma lahko pokrival tudi druge predmete, vendar bi ga bilo verjetno treba zaradi njihove specifičnosti še prilagoditi. Tukaj imamo v mislih predvsem matematiko, kjer bi morali zagotoviti še podporo za pisanje matematičnih izrazov in možnost, da bi sistem na koncu sam izračunal rezultat in ga primerjal z rezultatom, ki ga je vnesel učenec. Podobno bi lahko rekli tudi za kemijske reakcije in vse ostale specifičnosti.

Težava, ki je povezana s tem in je prisotna v slovenskem šolstvu, je, da se računalniki, razen pri informatiki, pri pouku ne uporabljajo oziroma je to zaenkrat prisotno redkokje. Ravno tako imajo po navadi šole samo eno ali dve učilnici z računalniki. Zaradi tega bi sistem lahko uporabljali zgolj doma ali pri pouku informatike.

5.1 Nadaljnje delo

Vsekakor bi morali preveriti, ali smo z našim sistemom enako uspešni tudi pri ostalih predmetih ali pa bi bilo pri nekaterih boljše, če ga ne bi uporabljali. Glede na učenčev učni slog bi se ga dalo razširiti z zelo naprednim prilagajanjem. Da bi dobili še boljši rezultat, bi morali preizkusiti še druge načine priporočanja.

V okviru našega dela se nismo posvetili prilagoditvam učnega sistema učenecem s posebnimi potrebami. To bi bilo zelo smiselno, še posebej, če bi želeli sistem uporabljati zelo na široko. Poleg tega sistem posameznemu učenecu ne da nobene informacije o tem, kje je v primerjavi z drugimi učenci. Enako velja tudi za razlago posameznih pravilnih odgovorov pri nalogah. Priporočanje in naloge bi bilo mogoče nadgraditi tako, da bi upoštevali različne taksonomske ravni.

Naše rezultate bi bilo vsekakor potrebno izboljšati, da dosežemo boljšo uspešnost od naključnega priporočanja. To bi lahko naredili z uporabo drugih algoritmov za priporočanje, ki dobro delujejo še posebno pri majhnih naborih podatkov. Vsekakor bi morali povečati tudi število uporabnikov in gradiv. Poiskati bi morali tudi pomanjkljivosti v naši učni množici.

Tudi na področju učne analitike pričakujemo še veliko novih idej in tukaj so odprte še številne možnosti za pridobivanje pomembnih podatkov o napredku učenca. Vse to je treba učitelju prikazati strnjeno in ga opozoriti na posamezne težave, ki jih ima določeni učenec. Določenih podatkov sedaj sploh še ne moremo dobiti, saj se e-učni sistemi še ne uporabljajo pri vseh predmetih – določenih težav tako še ne moremo zaznati.

Literatura

- [1] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, H. Thüs, A reference model for learning analytics, *International Journal of Technology Enhanced Learning* 4 (5-6) (2012) 318–331.
- [2] M. Kesim, H. Altınpulluk, The future of lms and personal learning environments, *Procedia Social and Behavioral Sciences* (2013) 1–5.
- [3] O. Bohl, J. Scheuhase, R. Sengler, U. Winand, The sharable content object reference model (scorm)-a critical review, in: *Computers in education, 2002. proceedings. international conference on, IEEE, 2002*, pp. 950–951.
- [4] T. R. McLain, Learning management systems adoption conundrums; technological and pedagogical dilemmas that arise for higher education, *Journal of Education and Training* 4 (2) (2017) 124–130.
- [5] R. S. Baker, K. Yacef, The state of educational data mining in 2009: A review and future visions, *JEDM-Journal of Educational Data Mining* 1 (1) (2009) 3–17.
- [6] M. Salehi, I. N. Kamalabadi, Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree, *Knowledge-Based Systems* 48 (2013) 57–69.
- [7] T. Arh, V. Rajkovič, B. Jerman-Blažič, Tehnološko podprto izobraževanje–uporabnost in primernost sistemov za upravljanje e-

- izobraževanja, Vzgoja in izobraževanje v informacijski družbi. Kranj: Moderna organizacija (2005) 386–393.
- [8] I. Balaban, G. Bubas, Educational potentials of eportfolio systems: Student evaluations of mahara and elgg, in: Information Technology Interfaces (ITI), 2010 32nd International Conference on, IEEE, 2010, pp. 329–336.
- [9] T. Boyle, Design principles for authoring dynamic, reusable learning objects, Australian Journal of Educational Technology 19 (1) (2003) 46–58.
- [10] Standards, dostopno na: <https://docs.moodle.org/28/en/Standards>, (pridobljeno: 11.9.2017).
- [11] M. Rey-López, R. P. Díaz-Redondo, A. Fernández-Vilas, J. J. Pazos-Arias, J. García-Duque, A. Gil-Solla, M. Ramos-Cabrer, An extension to the adl scorm standard to support adaptivity: The t-learning case-study, Computer Standards & Interfaces 31 (2) (2009) 309–318.
- [12] A. L. Dyckhoff, D. Zielke, M. Bültmann, M. A. Chatti, U. Schroeder, Design and implementation of a learning analytics toolkit for teachers., Educational Technology & Society 15 (3) (2012) 58–76.
- [13] G. Siemens, D. Gasevic, C. Haythornthwaite, S. P. Dawson, S. Shum, R. Ferguson, E. Duval, K. Verbert, R. Baker, et al., Open learning analytics: an integrated & modularized platform.
- [14] L. France, J.-M. Heraud, J.-C. Marty, T. Carron, J. Heili, Monitoring virtual classroom: Visualization techniques to observe student activities in an e-learning system, in: null, IEEE, 2006, pp. 716–720.
- [15] C.-M. Chen, H.-M. Lee, Y.-H. Chen, Personalized e-learning system using item response theory, Computers & Education 44 (3) (2005) 237–255.

-
- [16] M. Salehi, I. N. Kmalabadi, A hybrid attribute-based recommender system for e-learning material recommendation, *IERI Procedia* 2 (2012) 565–570.
 - [17] T.-H. Wang, Developing an assessment-centered e-learning system for improving student learning effectiveness, *Computers & Education* 73 (2014) 189–203.
 - [18] L. Sunil, D. K. Saini, Design of a recommender system for web based learning, in: *Proceedings of the World Congress on Engineering*, Vol. 1, 2013.
 - [19] B. Vesin, A. Klačnja-Milićević, M. Ivanović, Z. Budimac, Applying recommender systems and adaptive hypermedia for e-learning personalization, *Computing and Informatics* 32 (3) (2013) 629–659.
 - [20] T. Kaewkiriya, N. Utakrit, M. Tiantong, The design of a rule base for an e-learning recommendation system base on multiple intelligences, *International Journal of Information and Education Technology* 6 (3) (2016) 206.
 - [21] A. Klačnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, E-learning personalization based on hybrid recommendation strategy and learning style identification, *Computers & Education* 56 (3) (2011) 885–899.
 - [22] S. Graf, T.-C. Liu, et al., Identifying learning styles in learning management systems by using indications from students' behaviour, in: *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, IEEE, 2008, pp. 482–486.
 - [23] K. I. Ghauth, N. A. Abdullah, Learning materials recommendation using good learners' ratings and content-based filtering, *Educational technology research and development* 58 (6) (2010) 711–727.
 - [24] M. K. Khribi, M. Jemni, O. Nasraoui, Automatic recommendations for e-learning personalization based on web usage mining techniques

- and information retrieval, in: *Advanced Learning Technologies*, 2008. ICALT'08. Eighth IEEE International Conference on, IEEE, 2008, pp. 241–245.
- [25] I. Dimopoulos, O. Petropoulou, M. Boloudakis, S. Retalis, Using learning analytics in moodle for assessing students' performance.
- [26] D. Šajne, T. Urbančič, I. Arčon, Pregled in analiza nastajanja, preizkušanja in uporabe e-grafov pri pouku a review and analysis of the formation, testing and usage of e-materials in teaching, *Vzgoja in izobraževanje v informacijski družbi* (2009) 433–444.
- [27] Učni načrt informatika, dostopno na: http://portal.mss.edus.si/msswww/programi2011/programi/media/pdf/ucni_nacrti/UN_INFORMATIKA_gimn.pdf, (pridobljeno: 4.7.2017).
- [28] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, *Journal of machine learning research* 10 (Mar) (2009) 623–656.
- [29] H. Avancini, L. Candela, U. Straccia, Recommenders in a personalized, collaborative digital library environment, *Journal of Intelligent Information Systems* 28 (3) (2007) 253–283.
- [30] L. Zhuhadar, O. Nasraoui, R. Wyatt, E. Romero, Multi-model ontology-based hybrid recommender system in e-learning domain, in: *Web Intelligence and Intelligent Agent Technologies*, 2009. WI-IAT'09. IEEE/WI-C/ACM International Joint Conferences on, Vol. 3, IEEE, 2009, pp. 91–95.
- [31] H. Chau, J. Barria-Pineda, P. Brusilovsky, Content wizard: Concept-based recommender system for instructors of programming courses, in: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, ACM, 2017, pp. 135–140.

-
- [32] S. Cerón-Figueroa, I. López-Yáñez, W. Alhalabi, O. Camacho-Nieto, Y. Villuendas-Rey, M. Aldape-Pérez, C. Yáñez-Márquez, Instance-based ontology matching for e-learning material using an associative pattern classifier, *Computers in Human Behavior* 69 (2017) 218–225.
 - [33] J. Bernard, T.-W. Chang, E. Popescu, S. Graf, Learning style identifier: Improving the precision of learning style identification through computational intelligence algorithms, *Expert Systems with Applications* 75 (2017) 94–108.